STA450A

XMRADIO® SDARS SERVICE LAYER & SOURCE DECODER

FUNCTIONS

- XMRADIO SERVICE LAYER DEMULTIPLEXING
- CHANNEL AUXILIARY DATA MANAGEMENT
- CONDITIONAL ACCESS CONTROL
- SERVICE COMPONENTS EXTRACTION
- SERVICE COMPONENTS DECRYPTION
- HIGH QUALITY AUDIO DECODING WITH DATA RATE FROM 24kbps TO 96kbps
- HIGH QUALITY SPEECH DECODING WITH DATA RATE FROM 4kbps TO 16kbps
- 44.1 AND 32KHz SAMPLING FREQUENCIES SUPPORTED
- SYNCHRONIZATION ERROR DETECTION WITH SW INDICATORS

PERIPHERALS

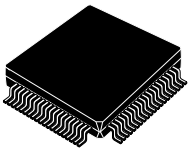
- HIGH SPEED SERIAL INPUT INTERFACE (PCBITSTREAM INTERFACE)
- FULLY PROGRAMMABLE SERIAL PCM OUTPUT INTERFACE
- IEC958 OUTPUT (S/PDIF)
- DATA OUTPUT PORT INTERFACE
- CONDITIONAL ACCESS PROCESSOR INTERFACE (I2C MASTER)
- RS232 RX & TX INTERFACES
- EMBEDDED SYSTEM PLL
- EMBEDDED AUDIO PLL

LOW POWER TECHNOLOGY

- 1.8V 0.18µm TECHNOLOGY
- 3.3V CAPABLE I/Os

CONTROL

- I2C SLAVE CONTROL
- DEVICE ADDRESS: 1011100



TQFP80
ORDERING NUMBER: STA450A

THIS DEVICE CAN BE SOLD ONLY TO CUSTOMERS THAT HAVE SIGNED A LICENSE AGREEMENT WITH XM SATELLITE RADIO.

DESCRIPTION

The STA450A is designed for digital radio receivers compatible with the XMRadio SDARS System and integrates all the functions needed to perform the Service Layer and Source Decoding:

- Bitstream Synchronization
- Service Layer (SL) Demultiplexing
- Auxiliary Data Management
- Conditional Access (CA) Control
- Service Components Extraction
- Service Components Decryption
- Audio and Voice Decoding

The extracted Audio and Data are made available through different interfaces:

- I2S Audio Output
- S/PDIF Output
- Data Output Port

Figure 1. Service Layer and Source Decoder Block Diagram

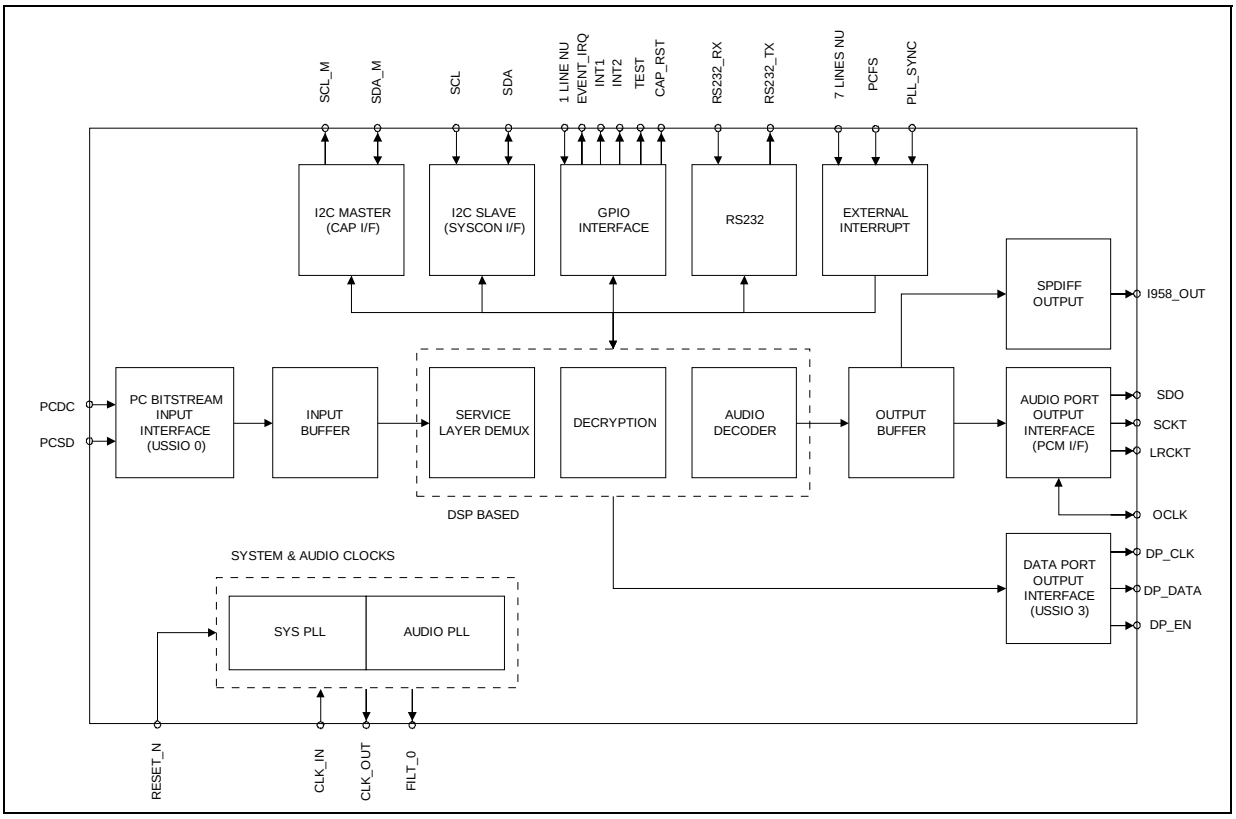
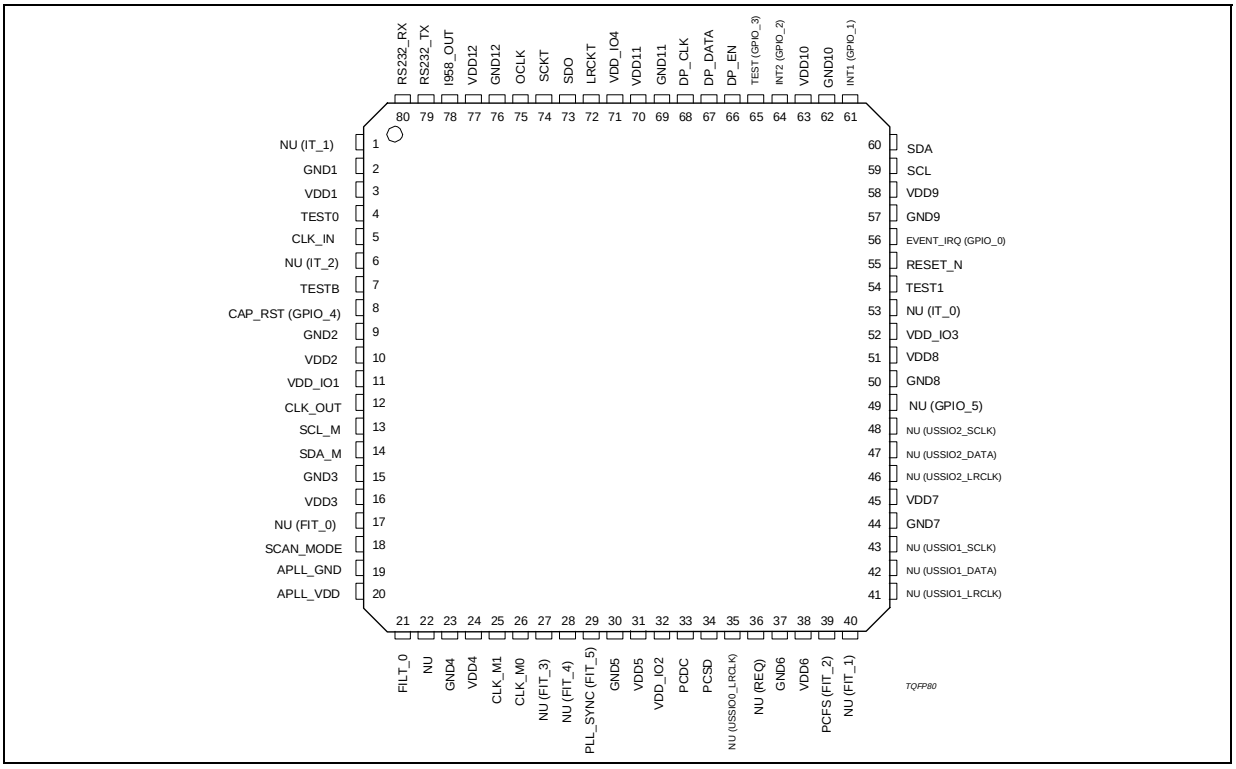


Figure 2. Pin Connection



PIN FUNCTION

N°	Pin	Type	Function	Pad Description
1,6,53	NU(IT_1), NU(IT_2), NU(IT_0)	I	Not Used (must be connected to ground in functional mode)	Schmitt Trigger Buffer
2,9,15,23, 30,37,44,5 0,57,62,69 ,76	GND1…,GND12		Ground	
3,10,16, 24,31,38,4 5,51,58,63 ,70,77	VDD1…,VDD12		1.8V Core Supply Voltage	
4	TEST0	I	Test Pin (must be connected to ground in functional mode)	Analog Pad
5	CLK_IN	I	Clock Input	Schmitt Trigger Buffer
7	TESTB	I	Test Pin	Input Pad Buffer with Active Pull-Up
8	CAP_RST (GPIO_4)	I/O	CAP Reset	Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
11,32,52,7 1	VDD_IO1,… VDD_IO4		3.3V I/O Supply Voltage	
12	CLK_OUT	O	Clock Out	4mA Output Buffer with Slew Rate Control
13	SCL_M	O	I2C Master Serial Clock	4mA Output Buffer with Slew Rate Control
14	SDA_M	I/O	I2C Master Serial Data	Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
17,27,28,4 0	NU(FIT_0), NU(FIT_3), NU(FIT_4), NU(FIT_1)	I	Not Used (must be connected to ground in functional mode)	Schmitt Trigger Buffer
18	SCAN_MODE	I	Scan Mode (must be connected to ground in functional mode)	Input Pad Buffer
19	APLL_GND		Analog Ground for Audio PLL	
20	APLL_VDD		1.8V Analog Supply for Audio PLL	
21	FILT_0		PLL Filter	Analog Pad
22	NU		Not Used	
25,26	CLK_M1, CLK_M0	I	Selection of Input Clock for the DSP core and peripherals	Input Pad Buffer
29	PLL_SYNC (FIT_5)	I	Fractional Audio PLL Sync.	Schmitt Trigger Buffer
33	PCDC	I/O	PC Bitstream Data Clock	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
34	PCSD	I/O	PC Bitstream Serial Data	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
35,41,46	NU(USSIO0_LRCLK) NU(USSIO1_LRCLK) NU(USSIO2_LRCLK)	I/O	Not Used (must be connected to ground in functional mode)	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
36	NU(REQ)	O	Not used	4mA Output Buffer with Slew Rate Control

PIN FUNCTION (continued)

N°	Pin	Type	Function	Pad Description
39	PCFS(FIT_2)	I	PC Bitstream PRC Frame Sync.	Schmitt Trigger Buffer
42,47	NU(USSIO1_DATA) NU(USSIO2_DATA)	I/O	Not Used (must be connected to ground in functional mode)	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
43,48	NU(USSIO1_SCLK) NU(USSIO2_SCLK)	I/O	Not Used (must be connected to ground in functional mode)	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
49	NU(GPIO_5)	I/O	Not Used (must be connected to ground in functional mode)	Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
54	TEST1	I	Test Pin	Input Pad Buffer with Active Pull-Up
55	RESET_N	I	HW Reset (active low)	Schmitt Trigger Buffer
56	EVENT_IRQ (GPIO_0)	I/O	General Interrupt (Events, Errors, Data ready)	Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
59	SCL	I	I2C Slave Serial Clock	Schmitt Trigger Buffer
60	SDA	I/O	I2C Slave Serial Data	BiDir Pad, 4mA Output Buffer with Slew Rate Control
61	INT1 (GPIO_1)	I/O	Dedicated Errors Interrupt	Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
64	INT2 (GPIO_2)	I/O	Dedicates Events Interrupt	Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
65	TEST (GPIO_3)	I/O	Test Pin	Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
66	DP_EN	I/O	Data Port Enable	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
67	DP_DATA	I/O	Data Port Data	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
68	DP_CLK	I/O	Data Port Clock	Schmitt Trigger Bi Dir Pad, 4mA Output Buffer with Slew Rate Control
72	LRCKT	O	I2S Left&Right Clock	4mA Output Buffer with Slew Rate Control
73	SDO	O	I2S Output Data (PCM Data)	4mA Output Buffer with Slew Rate Control
74	SCKT	O	I2S Serial Clock	4mA Output Buffer with Slew Rate Control
75	OCLK	I/O	Oversampling Clock	Schmitt Trigger BiDir Pad, 4mA Output Buffer with Slew Rate Control
78	I958_OUT	O	S/PDIF Output	4mA Output Buffer with Slew Rate Control
79	RS232_TX	O	RS232 Transmitter	4mA Output Buffer with Slew Rate Control
80	RS232_RX	I	RS232 Receiver (must be connected to ground in functional mode)	Schmitt Trigger Buffer

A more detailed description of each pad type can be found in section 4.

Note: After an Hardware reset or power on all Bi Dir pins are configured as inputs.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	1.8V Power supply Voltage	-0.5 to 2.5	V
V_{DD_IO}	3.3V Power Supply Voltage	-0.5 to 4	V
APLL_V_{DD}	1.8V Analog Supply Voltage	-0.5 to 2.5	V
V_i	Voltage on input pin	-0.5 to (V_{DD_IO} + 0.5)	V
V_o	Voltage on output pin	-0.5 to (V_{DD_IO} + 0.5)	V
T_{stg}	Storage temperature	-55 to +150	ºC
T_{oper}	Operative Ambient Temperature	-40 to +85	ºC
T_j	Operative Junction Temperature	-40 to +125	ºC

THERMAL DATA

Symbol	Parameter	Value	Unit
$R_{j\text{-}amb}$	Thermal Resistance Junction to Ambient (1)	40	°C/W

(1) According to JEDEC specification on a 4 layers board

DC ELECTRICAL CHARACTERISTCS (T_{amb} = -40 to 85°C, V_{DD} = APLL_V_{DD} = 1.65 to 1.95V, V_{DD_IO} = 3.0 to 3.6V unless otherwise specified).

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	1.8V Supply Voltage		1.65	1.8	1.95	V
V_{DD_IO}	3.3V Supply Voltage		3.0	3.3	3.6	V
APLL_V_{DD}	1.8V Analog Supply Voltage		1.65	1.8	1.95	V
I_{DD}	V_{DD} Power Supply Current	f_{CLK} = 23.92MHz; 56kbps audio stream; V_{DD}=1.95V		110	140	mA
I_{DD_IO}	V_{DD_IO} Power Supply Current	f_{CLK} = 23.92MHz; 56kbps audio stream; V_{DD_IO}=3.6V; V_{DD}=1.95V		24	40	mA
I_{DD_APLL}	APLL_V_{DD} Power Supply Current	f_{CLK} = 23.92MHz; APLL_V_{DD} =1.95V		0.5	1.2	mA
P_D	Power Dissipation	V_{DD} = 1.8V; V_{DD_IO} = 3.3V		220		mW
I_{il}	Low level input leakage current 4)	Vi = 0V			1	µA
I_{ih}	High level input leakage current 4)	Vi = V_{DD3}			1	µA
I_{OZ}	Tristare output leakage current 5)	Vo = 0V or V_{DD3}			1	µA
I_{pu}	Pull-up current	Vi = 0V	40		120	µA
R_{pu}	Equivalent pull-up resistance 3)	Vi = 0V		50		kΩ
V_{il}	Low level input voltage				0.8	V
V_{ih}	High level input voltage		2			V
V_{ilhyst}	Low level threshold input falling		0.8		1.35	V
V_{ihhyst}	High level threshold input rising		1.3		2	V
V_{hyst}	Schmitt trigger hysteresy 1)		0.3		0.8	V

DC ELECTRICAL CHARACTERISTCS (continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{ol}	Low level output voltage 2)	Iol = 4mA			0.2	V
V_{oh}	High level output voltage 2)	Ioh = 4mA	2.8			V
C_{IN}	Input Capacitance 1)				1.2	pF
C_{OUT}	Output Capacitance 1)				1.9	pF
C_{IO}	I/O (BiDir) Capacitance 1)				2.1	pF
$I_{latchup}$	I/O Latching Current		200			mA
V_{ESD}	Electrostatic Protection	Leakage<1µA	4000			V

Note 1. Guaranteed by Design
Note 2. Take into account 200 mV voltage drop in supply lines and Input/Output levels for frequency > 20MHz.
Note 3. Guaranteed by Ipu measurements
Note 4: Performed on all the input pins excluded the pull-down ones
Note 5: Performed on the I/O pins in tristate mode

1.0 FUNCTIONAL DESCRIPTION

1.1 PC BITSTREAM INTERFACE

The STA450A receives a serial data stream from the STA400A Channel Decoder via the PC Bitstream Interface.

Figure 3 shows the stream format outputs from the CDEC (STA400A).

Figure 3. PC Bitstream Interface protocol

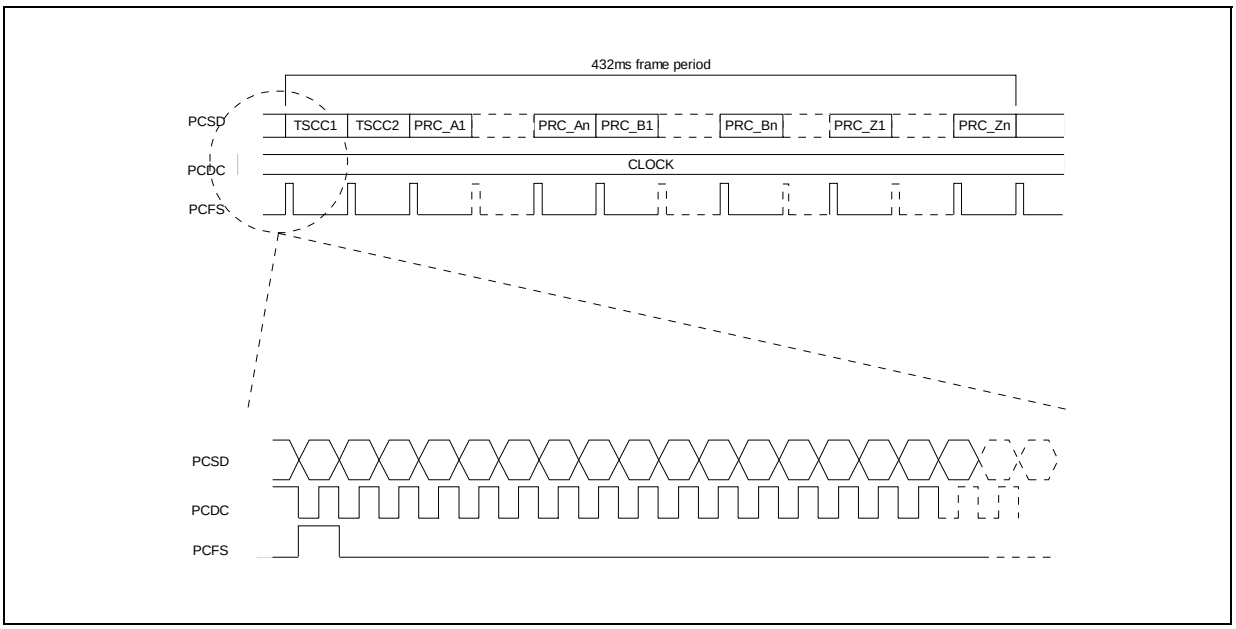


Table 1 shows the correspondence between CDEC output clock divider register and the data rate on the bus

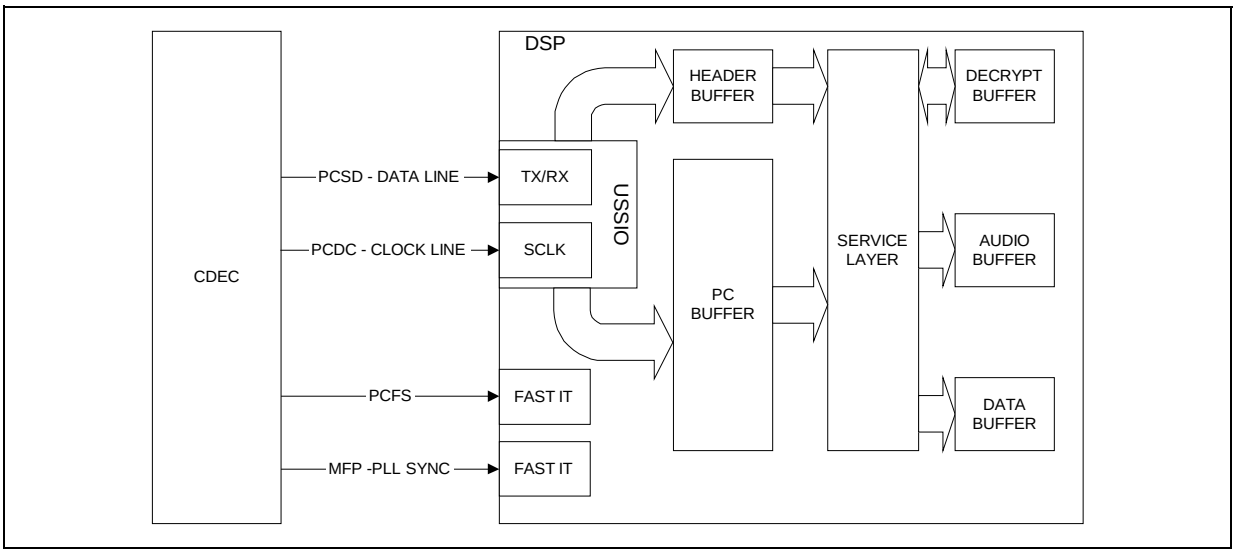
Table 1. Suggested CDEC clock divider/output rate correspondence.

MCLK (Hz)	Divider	F Out (Hz)	T Out (sec)	T Burst (sec)
23920000	2	11960000	8.361E-08	2.997E-04
	4	5980000	1.672E-07	5.993E-04
	6	3986667	2.508E-07	8.990E-04
	8	2990000	3.344E-07	1.199E-03

The 432ms frame is divided in slots of 448 bytes; the minimum number af slots in a 432ms frame is 50, corresponding to a Tslot of 8.64ms. The transmission is on bursts, whose duration can go from 299us up to 1.199ms (refer to table 1) according to the clock divider factor selected in the STA400A.

An overview of the connection between the STA400A and the STA450A is given in Figure 4.

Figure 4. STA400A/STA450A Connection.



The synchronization process is started after a PCFS pulse is received through a Fast Interrupt port (pin 39). In the case the PC Bitstream is stopped or the clock line is perturbed, the PCFS signal is used to reset the firmware State Machine in the PC Bitsteam Interface (STA450A); this mechanism guarantees the start of the resync process from a stable state.

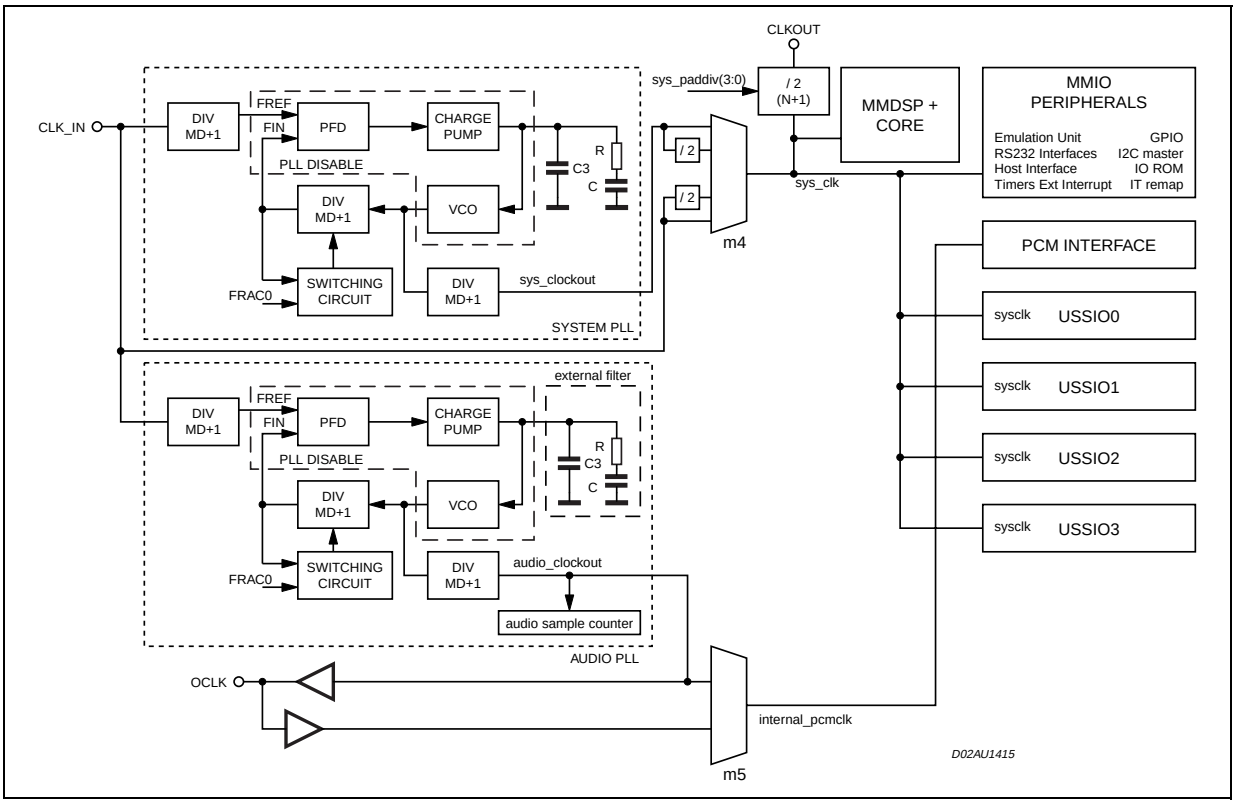
The PC Bitstream Interface receives the data and fills the Header and the PC buffers; the Header buffer contains the Header byte 1, the Header byte 2 and the Service preamble, the PC buffer contains the Payload channels data.

1.2 STA450A Clocks Generation System

The STA450A Clocks generation System is shown in Figure 5: the 23.92MHz reference signal is entered in the pin CLK_IN and is used to generate an internal system clock, with a System PLL, and an internal PCM clock, with an Audio Fractional PLL .

A clock is made available at the output through the pin CLK_OUT.

Figure 5. STA450A Clocks Generation System



The system clock

The system clock sent to the DSP core and the peripherals can be derived from 4 sources of clock and the selection is performed by m4. The control of m4 is performed by the 2 input pins (CLK_M1 & CLK_M0):

- 00 sys_clk = CLK_IN
- 01 sys_clk = CLK_IN divided by 2
- 10 sys_clk = sys_clockout
- 11sys_clk = sys_clockout divided by 2

In the chip, the duty cycle of the internal system clock must be as close as possible 50%.

When the sys_clk is derived from the external clock source (CLK_IN), the divider by 2 can be used to ensure a 50% duty cycle.

When the sys_clk is derived from the system PLL, the duty cycle of the sys_clockout signal is 50%. The divider by 2 can be used to slow down the clock at a lower frequency than the minimum that can achieve the System PLL.

The CLK_OUT pad is driven by the sys_clk divided by a programmable division factor ranging from 1 to 16; the sys_clk frequency is divided by 2 * (N+1), where N is the value programmed with register sys_paddiv [3:0].

Configuration Equations for the System PLL:

$$F_{out} = \frac{1}{X+1} \cdot \frac{Fref_{IN}}{N+1} \cdot \left(M + 1 + \frac{reference}{65536}\right)$$

X = pllsys_Xmodulo

N = pllsys_Nmodulo

M = pllsys_Mmodulo

The System PLL registers are configured through an indirection mechanism using the HOST_pll_add, HOST_pll_data and HOST_pll_cmd registers; the HOST_pll_cmd register allows to update the control registers of the PLL.

There are two levels of registers (Level 1 & Level 2). The first level of registers (Level 1) is configured through the indirection mechanism. The second level of registers (Level 2) is a copy of the previous level in order to update all the configuration bits at the same time. This mechanism avoids to have, during the configuration phase, intermediate configurations that are not in line with the final desired configuration.

Assuming a 23.92MHz CLK input frequncy and CLK_M[1:0] = “11”, the default system frequency is 59.8MHz

See also the register description.

The Audio PLL

The "internal_pcmclk" of STA450A can be provided by two different sources: the Audio PLL or the OCLK port of the chip.

The "m5" multiplexor and the direction of the OCLK tri-state port are configured by the register HOST_Pllpcm (address 0x12).

An Audio PLL is embedded in STA450A.

The particularity of STA450A Audio PLL is the possibility to modify the Audio Sampling Frequency (LRCKT) in steps of a few p.p.m. to compensate dynamically the audio sampling frequency offset between the receiver and the broadcasting station; this compensation produces a jittering effect outside the audible range.

The STA450A receives from the STA400A (Channel Decoder) a dedicated signal every 432ms (PLL_SYNC) and uses this signal to perform the audio sampling rate compensation; the control is done by the DSP core updating the internal PLL registers.

Some PLL configuration registers are made available to the user to configure the PCM output according to the used DAC.

The programmation for the desired Fs should be accomplished for both the Fs = 48KHz and Fs = 44.1KHz families before the start-up of the DSP (write in the register 0 x 4D)

The OCLK frequency can be derived from the following formula:

$$OCLK\_freq = \frac{1}{1+X} \cdot \frac{23.92MHz}{1+N} \cdot \left(M + 1 + \frac{FRAC}{65536}\right)$$

- X is the value of the HOST_APLL48_XDIV register (HOST_APLL441_XDIV) register.
- M is the value of the HOST_APLL48_MDIV register (HOST_APLL441_MDIV) register.
- N is the value of the HOST_APLL48_NDIV register (HOST_APLL441_NDIV) register.
- FRAC is the decimal value of the concatenated registers HOST_APLL48_LSB and HOST_APLL48_MSB (HOST_APLL441_LSB and HOST_APLL441_MSB) as follows:
  FRAC = 256 * HOST_APLL48_MSB + HOST_APLL48_LSB (= 256 * HOST_APLL441_MSB + HOST_APLL441_LSB)

The changes in the registers are not effective once the DSP has been started.

According to the chosen oversampling factor, the following table permits to configure the dividers of the Audio PLL:

O_FAC	N	M	FRAC	X
Fs=48KHz family				
256	1	15	28756	15
384	1	14	26959	9
512	1	15	28756	7
Fs=44.1KHz family				
256	1	15	64948	17
384 (default)	1	14	37691	10
512	1	15	64948	8

See also the register description.

1.3 PCM Output Interface

The decoded audio data can be output in serial PCM format.

The interface consists of the following signals:

SDO PCM Serial data output

SCKT PCM serial Clock Output

LRCKT Left/Right Channel Selection Clock

The output samples precision is selectable from 16 to 24 bits/word by setting the output precision (16, 18, 20 and 24bits) with the HOST_PCMCNF register; the same register is used to output data either with the most significant bit first (MS) or least significant bit first (LS).

Figure 6 gives a description of the STA450A PCM Output Formats.

The SCKT signal is the bit clock for the serial output and is derived from the PCMCLK (OCLK) as in the following formula:

SCKT = OCLK / 2*(HOST_PCMDIV + 1)

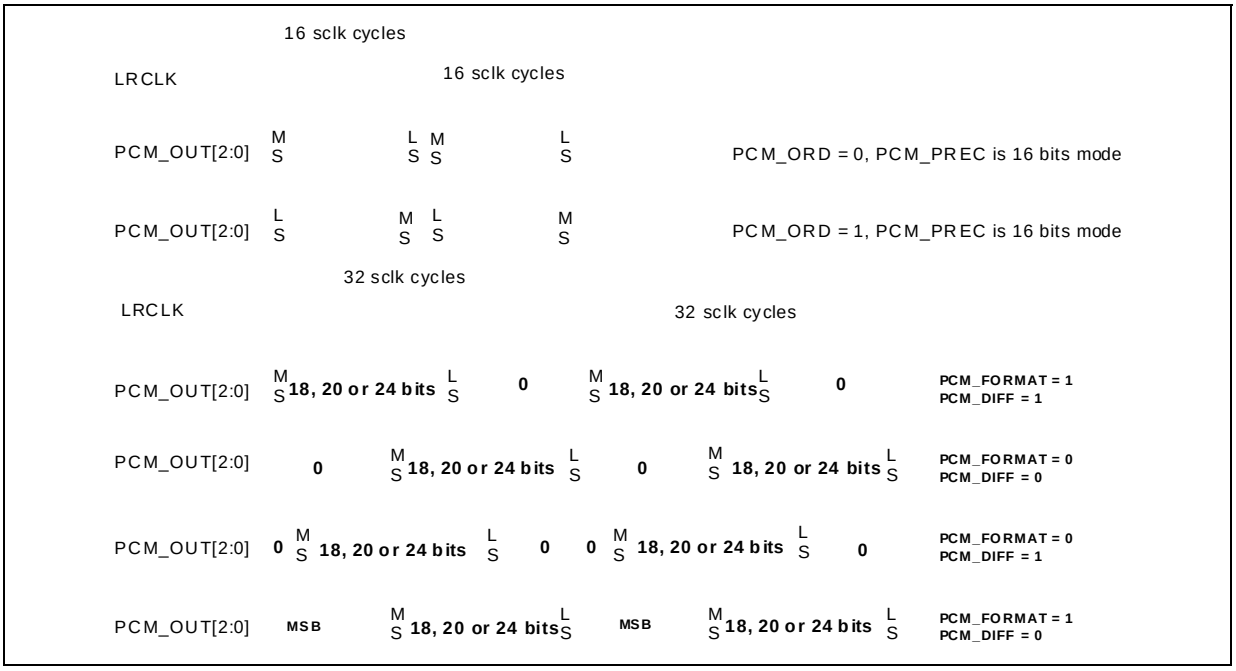
The number of bits to be transmitted to the DAC during one LRCKT clock period depends on the DAC precision (16, 18, 20 or 24bits) and on the mode used to transmit the data (LRCKT_period equal to 16x2 or 32x2 SCKT_period - refer to figure 6).

The value of the HOST_PCMDIV register must be set accordingly to the previous consideration and to the DAC Oversampling Factor (O_FAC).

– LRCKT_period = 16x2 SCKT_period
  HOST_PCMDIV = (O_FAC/64) - 1
– LRCKT_period = 32x2 SCKT_period
  HOST_PCMDIV = (O_FAC/128) - 1

See also the register description.

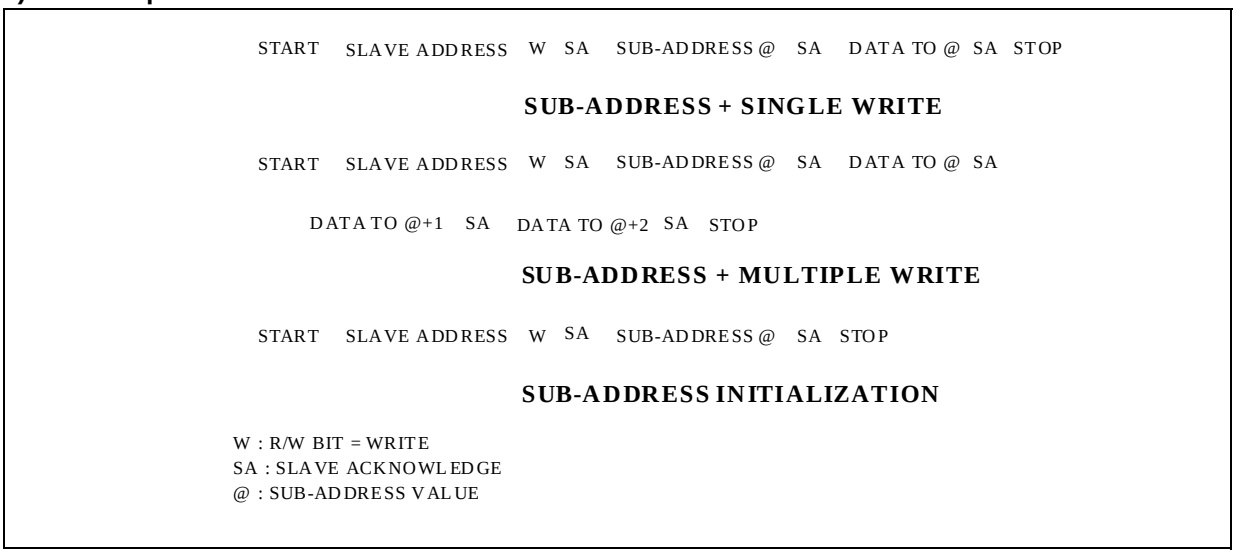
Figure 6. PCM Output Formats



1.4 Master Interface I2C

The I2C master interface is used by the STA450A to communicate with the external Conditional Access Processor (CAP) with the following protocol:

a) Write sequences



b) Read sequences

START SLAVE AD- R SA DATA NMA STOP

SINGLE READ

START SLAVE ADDRESS R SA FIRST DATA MA SECOND DATA NMA STOP

MULTIPLE READ

START SLAVE ADDRESS W' SA SUB-ADDRESS SA STAR SLAVE AD- R SA DATA FROM @ NMA STOP

COMBINED FORMAT : SUB-ADDRESS + SINGLE READ

START SLAVE ADDRESS W SA SUB-ADDRESS @ SA START SLAVE ADDRESS R SA DATA FROM @ MA

DATA FROM @ + 1 NMA STOP

COMBINED FORMAT : SUB-ADDRESS + MULTIPLE READ

R : R/W BIT = READ
SA : SLAVE ACKNOWLEDGE
MA : MASTER ACKNOWLEDGE
NMA : NO MASTER ACKNOWLEDGE (LAST DATA)
@ : SUB-ADDRESS VALUE

1.5 Data Output Port

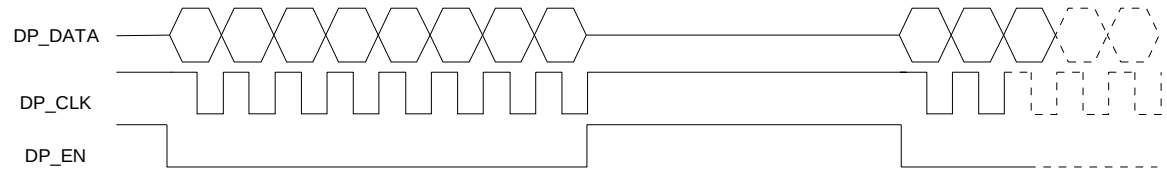
The STA450A sends data through the Data Port. The Data Port consists of 3 lines: DP_CLK, DP_DATA, DP_EN. The communication protocol is in burst.

The data changes on the DP_DATA line on the raising edge of the clock line DP_CLK, the DP_EN line defines when the data is valid; DP_DATA and DP_EN must be sampled on the falling edge of DP_CLK.

The Service Component is output through the Data Port without adding a flag indicating the frame.

The Data Port burst rate is selectable through the XM Stack Command DATAPORT_CH_FREQUENCY (op-code 0x07) from 1 MHz to 12 MHz, the default rate is 2 MHz.

Figure 7. Data Port Protocol Waveform Diagram



1.6 IEC958 Output (SPDIF)

The SPDIF output is a Fully IEC958 formatted, single ended output for linear PCM output (left and right channel, 16,18, 20 & 24 bits) supporting the Consumer Mode.

1.7 Test Interfaces (RS232)

The STA450A provides a RS232 RX and a RS232TX interfaces for testing purposes.

2.0 I2C BUS SPECIFICATION

The STA450A supports the I2C protocol to communicate with the System Controller; the STA450A is always a slave in its communication to the System Controller.

2.1 COMMUNICATION PROTOCOL

A data change on the SDA line must only occur when SCLKI2C clock is low except for START and STOP conditions. In that case, the transition is done when the clock is High.

A START condition is identified by a High to Low transition of the SDA line while the clock signal is High. A START condition must precede any command for a data transfer.

A STOP condition is identified by a Low to High transition of the SDA line while the clock signal SCLKI2C is High. A STOP condition terminates the communications between the IC and the master of the I2C bus.

An Acknowledge bit is used to indicate a successful data transfer. The bus transmitter, either the master or the slave, releases the SDA line after sending 8 bits of data. During the 9th clock pulse, the receiver pulls the SDA line to Low to acknowledge the reception of 8 bits of data.

During the data transfer, the I2C slave interface of the IC samples the SDA line on the rising edge of the SCLKI2C clock. The SDA signal has to be stable during the rising edge of the clock and the data can change only when the SCLKI2C clock line is low.

2.2 DEVICE ADDRESSING

To start the communications between the master and the IC, the master must initiate the transfer with a START condition. Then, the master has to send on the SDA line 8 bits (MSB first) corresponding to the device I2C address (7 bits) and the mode bit RW (Read or Write).

The 7 most significant bits are the address of the device. For the STA450A the address is 0x5C (1011100 address on 7 bits).

The 8th bit (LSB) selects a read (bit set to 1) or write (bit set to 0) operation. After a START condition, the IC I2C slave interface identifies on the I2C bus the device address and, if the address matches, the IC acknowledges this match on the SDA line during the 9th bit time frame. The byte following the device identification byte is the address of the Host register to be accessed.

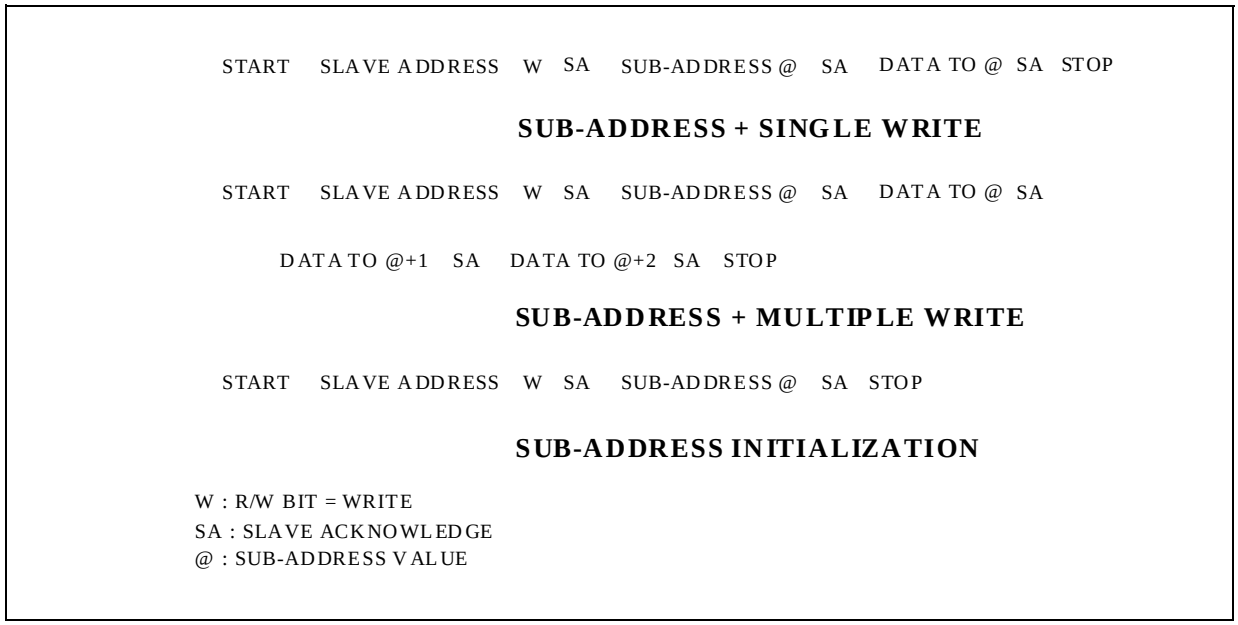
2.2.1 Sub-address initialization

This mode is used for the initialization of the Host address register (sub-address value). The Host address register is the register that points the data register to be accessed (read or write).

2.2.2 "Sub-address + single write" & "Sub-address + multiple write"

The second mode, the multiple write, exploits the autoincrementation of the sub-address pointer to avoid to initialize, for sequential accesses of the Host registers, the sub-address at each write operation. The length of a multiple write is limited to the size of the Host register area (256 locations).

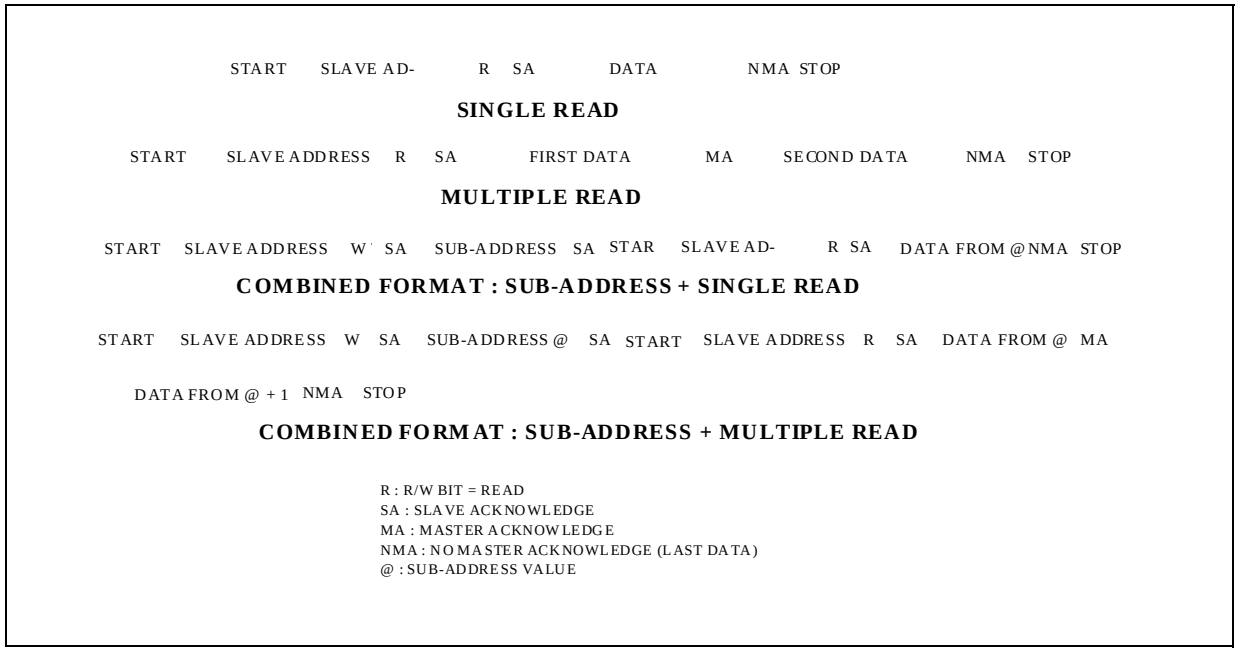
After a writing in the I2C interface a interrupt is generated to the core if the System controller set the bit in the HOST_Cmd0 register.



2.2.3 "Single read" & "multiple read" & "sub-address + single read" & "sub-address + multiple read"

The single read operations are performed from the current sub-address value. The sub-address value can be initialized using the "sub_address" initialization" sequence presented in the previous transfer chart : see the "combined format - sub-address + single read" diagram in the following chart.

The multiple read operations are performed from the current sub-address value and the sub-address register is automatically incremented at each access. The sub-address value can be initialized using the "sub_address initialization" sequence presented in the previous transfer chart : see the "combined format - sub-address + multiple read" diagram in the following chart. The length of a multiple read is limited to the size of the Host register area (256 locations). A multiple read is terminated by a Non Master Acknowledge followed by a STOP condition.



2.3 REGISTER MAP

The DSP HOST interface includes 256 registers.

HOST @	Name	Register size	Mode	SW Reset	HW Reset	Comment
0x00	HOST_VERSION	8 bits	R	0x10	0x10	
0x01	HOST_ID	8 bits	R	0x20	0x20	
0x02 to 0x0F	Reserved					
0x10	HOST_SOFTRESET	8 bits	W	NA	NA	Soft reset of the DSP core and peripherals
0x11	HOST_Plldata	8 bits	R – W	NC	0	Data register to configure the different configuration registers
0x12	HOST_Pllpcm	3 bits	R – W	NC	01	PCM clock direction configuration
0x13	HOST_cmd0	1 bit	R – W	0	0	I2C Interrupt request
0x14 to 0x17	Reserved					
0x18	HOST_Pllcmd	8 bits	R – W	NC	0	Command register to configure the PLLs
0x19 to 0x1B	Reserved					
0x1C	HOST_I2cdiv	6 bits	R – W	0x0B	0x0B	Hold time value of the data on SDA versus SCL edges
0x1D	HOST_Plladd	8 bits	R – W	NC	0	Address register to configure the different configuration registers
0x1E	HOST_SerialDivL	8 bits	R – W	0x00	0x00	RS232 rate coeff L
0x1F	HOST_SerialDivH	8 bits	R – W	0x00	0x00	RS232 rate coeff H
0x20 to 0x2A	Reserved					
0x2B	Host_Memory Access	4 bits	W	0x00	0x00	Enables core to access ucode
0x2C to 0x39	Reserved					
0x3A	Host_clkstop	1bit	W	NA	1	Stops the clock to the core
0x3B to 0x3F	Reserved					
0x40	HOST_SOFTVER	8 bits	R – W	NA	NA	Software version (BCD)
0x41	HOST_EVENTINTE0	8 bits	R – W	NA	NA	EVENT Interrupt enable bit[7:0]
0x42	HOST_EVENTINTE1	8 bits	R – W	NA	NA	EVENT Interrupt enable bit[15:8]
0x43	HOST_EVENTINTE2	8 bits	R – W	NA	NA	EVENT Interrupt enable bit[23:16]
0x44	HOST_EVENTINTE3	8 bits	R – W	NA	NA	EVENT Interrupt enable bit[31:24]
0x45	HOST_EVENTINT0	8 bits	R – W	NA	NA	EVENT Interrupt value bit[7:0]
0x46	HOST_EVENTINT1	8 bits	R – W	NA	NA	EVENT Interrupt value bit[15:8]

HOST @	Name	Register size	Mode	SW Reset	HW Reset	Comment
0x47	HOST_EVENTINT2	8 bits	R – W	NA	NA	EVENT Interrupt value bit[23:16]
0x48	HOST_EVENTINT3	8 bits	R – W	NA	NA	EVENT Interrupt value bit[31:24]
0x49	HOST_ERRINTEL	8 bits	R – W	NA	NA	ERROR Interrupt enable bit[7:0]
0x4A	HOST_ERRINTEH	8 bits	R – W	NA	NA	ERROR Interrupt enable bit[15:8]
0x4B	HOST_ERRINTL	8 bits	R – W	NA	NA	ERROR Interrupt value bit[7:0]
0x4C	HOST_ERRINTH	8 bits	R – W	NA	NA	ERROR Interrupt value bit[15:8]
0x4D	HOST_STARTUP	8 bits	R – W	NA	NA	Startup
0x4E	HOST_PCMDIV	8 bits	R – W	NA	NA	Pcm clock divider
0x4F	HOST_PCMCNF	8 bits	R – W	NA	NA	Pcm Configuration
0x50	HOST_APLL48_LSB (1)	8 bits	R – W	NA	NA	Fractional PLL LSB reference value for Fs = 48KHz family
0x51	HOST_APLL48_MSB(1)	8 bits	R – W	NA	NA	Fractional PLL MSB reference value for Fs = 48KHz family
0x52	HOST_APLL48_XDIV(1)	8 bits	R – W	NA	NA	Fractional PLL X Divider for Fs = 48KHz family
0x53	HOST_APLL48_MDIV(1)	8 bits	R – W	NA	NA	Fractional PLL M Divider for Fs = 48KHz family
0x54	HOST_APLL48_NDIV(1)	8 bits	R – W	NA	NA	Fractional PLL N Divider for Fs = 48KHz family
0x55	HOST_APLL441_LSB(1)	8 bits	R – W	NA	NA	Fractional PLL LSB reference value for Fs = 44.1KHz family
0x56	HOST_APLL441_MSB(1)	8 bits	R – W	NA	NA	Fractional PLL MSB reference value for Fs = 44.1KHz family
0x57	HOST_APLL441_XDIV(1)	8 bits	R – W	NA	NA	Fractional PLL X Divider for Fs = 44.1KHz family
0x58	HOST_APLL441_MDIV(1)	8 bits	R – W	NA	NA	Fractional PLL M Divider for Fs = 44.1KHz family
0x59	HOST_APLL441_NDIV(1)	8 bits	R – W	NA	NA	Fractional PLL N Divider for Fs = 44.1KHz family
0x5A	ENABLE_IT432	8 bits	R – W	NA	NA	Enable control with FRAC PLL
0x5B	IT432_CONF	8 bits	R – W	NA	NA	Edge configuration for IT432
0x5C	HOST_MaxDev	8 bits	R – W	NA	NA	Hax duration windows divider factor
0x5D	HOST_Decoder BitRate	8 bits	R – W	NA	NA	Active Audio decoder bitrate
0x5E	HOST_Bitstream Synchro	8 bits	R – W	NA	NA	Synchronization
0x5F	Host_DataPort_BitRate	8 bits	R – W	NA	NA	Data Port Bit Rate
0x60 to 0x65	Reserved					
0x66	HOST_EVENTINTE4	8 bits	R – W	NA	NA	EVENT interrupt enable bit [7:0]

HOST @	Name	Register size	Mode	SW Reset	HW Reset	Comment
0x67	HOST_EVENTINTE5	8 bits	R – W	NA	NA	EVENT interrupt enable bit [15:8]
0x68	HOST_EVENTINT4	8 bits	R – W	NA	NA	EVENT interrupt value bit [7:8]
0x69	HOST_EVENTINT5	8 bits	R – W	NA	NA	EVENT interrupt value bit [15:8]
0x6A	Reserved					
0x6B	AudioDec Result	8 bits	R – W	NA	NA	Audio Decoder Result
0x6C	HOST_MFC	8 bits	R – W	NA	NA	Master Frame counter (MFC) LSB part (7 bits)
0x6D to 0x7E	Command	8 bits	R – W	NA	NA	
0x7F	HOST_PAGE_CTRL	8 bits	R – W	NA	NA	More page indication and number of byte in the page
0x80 to 0xFF	Data Page	8 bits	R – W	NA	NA	

Note: NA: Not Applicable, NC: Not Change, R: Read Only, W: Write Only
Note1: These registers must be programmed before the start-up of the DSP (writing in register 0x4D)

2.4 REGISTER DESCRIPTION

HOST_VERSION

	7	6	5	4	3	2	1	0
00								

Address : 0x00
Type : R
Software Reset : 0x10
Hardware Reset : 0x10

Description
The VERSION register is read-only and is used to identify the IC cut . The VERSION register holds the cut number (binary decimal encoded)

HOST_ID

	7	6	5	4	3	2	1	0
01								

Address : 0x01
Type : R
Software Reset : 0x20
Hardware Reset : 0x20

Description
The HOST_ID register is read-only and is used to identify the IC on an application board. The ID is fixed for all IC cut. The STA450A has the ID 0xAC.

HOST_SOFTRESET

	7	6	5	4	3	2	1	0
10								

Address : 0x10
Type : W
Software Reset : NA
Hardware Reset : NA

Description
When bit 0 of this register is set, a soft reset occurs. The command registers and the interrupt registers are cleared. The STA450A goes into idle mode.

HOST_Pllpcm

	7	6	5	4	3	2	1	0
12								

Address : 0x12
Type : R - W
Software Reset : NC
Hardware Reset : 0x01
Description
00 Not used

01 The PCMCLK pad is in input. This external PCMCLK source is sent to the I2S PCM output.

10 The PCMCLK pad is in output. The internal audio PLL is generating the PCM clock for the I2S PCM output.
11 Same case as 10 but the PCMCLK is in tri-state mode

HOST_Plldata

	7	6	5	4	3	2	1	0
11								

Address : 0x11
Type : R/W
Software Reset : 0
Hardware Reset : 0

Description
Data to be copied in the Level 1 register for system PLL.
This register should be used in conjunction with registers HOST_Pllcmd and HOST_Plladd

HOST_Pllcmd

	7	6	5	4	3	2	1	0
18								

Address : 0x18
Type : R/W
Software Reset : 0

Hardware Reset : 0

Description

Bit [1:0] 00: no action is performed on the configuration registers of the Level 1.
01: Read action of the configuration registers. During this phase, the content of a selected (by HOST_pll_add) configuration register of the Level 1 is copied into the HOST_pll_data register.
10: Write action of the configuration registers. During this phase, the content of the HOST_pll_data
register is copied into a selected (by HOST_pll_add) configuration register of the Level 1.
11: do not use.

Bit 2 The bit controls the transfer of the data between the Level 1 and the Level 2 for the System PLL. When this bit is set, all the registers of the Level 1 (sys_ndiv, sys_pdiv, sys_setupH, sys_setupL, sys_enable) are copied into the registers of the Level 2 at the same time.
When this bit is cleared, all the Level 2 registers have a stable state independently of the Level 1 registers.

Bit3 Reserved

Bit 4 This bit must be used when switching from one System PLL configuration to the other one. This bit must be used in conjunction with the bit [2].

HOST_Plladd

	7	6	5	4	3	2	1	0
1D								

Address : 0x1D
Type : R/W
Software Reset : 0
Hardware Reset : 0

Description

In the follow table the description of the registers addressable by the HOST_Plladd to control the system PLL

Add	Name	Size	Mode	SW Reset	HW Reset	Comment
3	pllsys_disable	1	R - W	NC	0	System PLL disable control 0 : system PLL enabled 1 : system PLL disabled
4	pllsys_F_low	8	R - W	NC	0	8 low bits of Fractional value for system PLL
5	pllsys_F_high	8	R - W	NC	0	8 high bits of Fractional value for system PLL
6	pllsys_S	5	R - W	NC	2	S divider for system PLL
7	pllsys_N	4	R - W	NC	1	N divider for system PLL
8	pllsys_X	7	R - W	NC	0	X divider for system PLL
9	pllsys_M	5	R - W	NC	9	M divider for system PLL
10(0xA)	pllsys_update_frac	1	R - W	NC	0	Update Fractional value for system PLL
12(0xC)	pllsys_paddiv	4	R - W	NC	3	pad clock divider

HOST_Cmd0

	7	6	5	4	3	2	1	0
13								Cmd0

Address : 0x13
Type : R/W
Software Reset : 0
Hardware Reset : 0

Description

A write into the bits 0 of this register generates a interrupt to the DSP core

HOST_I2cdiv

	7	6	5	4	3	2	1	0
1C			Div [5:0]					

Address : 0x1C
Type : R/W
Software Reset : 0x0B
Hardware Reset : 0x0B

Description

Hold time = HOST_I2cdiv/fc where the fc is the DSP system frequency.

HOST_SerialDivH - HOST_SerialDivL

	7	6	5	4	3	2	1	0
1F	Div [15:8]							
1E	Div[7;0]							

Address : 0x1F - 0x1E
Type : R/W
Software Reset : 0x00(HOST_SerialDivH) - 0x00 (HOST_SerialDivL)
Hardware Reset : 0x00 (HOST_SerialDivH) - 0x00 (HOST_SerialDivL)

Description

These registers are used to specify the frequency division factor of the system clock for the RS232 interface used for the emulation; bit rate = fc/((HOST_SerialDivH << 8) | HOST_SerialDivL)

HOST_MemoryAccess

	7	6	5	4	3	2	1	0
2B								

Address : 0x2B
Type : W
Software Reset : NA
Hardware Reset : 0

Description

Setting the bit 3 the core is enabled to access the ucode inside the memory

HOST_ClkStop

	7	6	5	4	3	2	1	0
3A								

Address : 0x3A
Type : R
Software Reset : NA
Hardware Reset : 0

Description
Clearing the bit 0 the clock to the core is started.

HOST_SOFTVER

	7	6	5	4	3	2	1	0
40								

Address : 0x40
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
The SOFTVER register is the version of the microcode which is running on the device (BCD).
This register is updated just after a soft reset of the device.

HOST_EVENTINTE 0 -1 -2 -3

	7	6	5	4	3	2	1	0
44	INTE[31:24]							
43	INTE[23:16]							
42	INTE[15:8]							
41	INTE[7:0]							

Address : 0x44 - 0x41
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
These registers are associated to error condition inside the STA450A.
The STA450A contains a 32 bits interrupt register associated with 32 bits enable register.
A bit set in this register will enable the generation of an external interrupt on the interrupt line.
The interrupt associated with each bit is given in the register INT description.

HOST_EVENTINT 0 - 1 - 2 - 3

	7	6	5	4	3	2	1	0
48	INT[31:24]							
47	INT[23:16]							
46	INT[15:8]							
45	INT[7:0]							

Address : 0x48 - 0x45
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
These registers are associated to error condition inside the STA450A.

A bit set in this table indicates the corresponding event has been occurred. Whenever the corresponding bit in the INTE table has been set an external interrupt is generated.

To clear the bit a fast command has to be issued (see command description).

Name	Reg	Bit	Comment
BAC_Authorization	0x45	0x01	The radio is entered in maintenance mode
BAC_Deauthorization		0x02	The radio is entered in activation mode
UTC		0x04	Universal Time Code
Response		0x08	I2C page ready to read by system controller
CRB_Table_Complete		0x10	The SDEC have received the CRB last block indication with new sequence number
BIC_ServiceLabel_0		0x20	Received a new Service label for selection in location 0
BIC_ProgramTypeLabel_0		0x40	Received a new Program Type label for selection in location 0
BIC_ServiceSelection_0		0x80	Received a new Service Selection message for selection in location 0
BIC_SongArtistLabel_0	0x46	0x01	Received a new Song/Artist label for selection in location 0
BIC_ChannelRefernceLabel_0		0x02	Channel Reference Table changed for selection in location 0
ADF_ExtArtistLabel_0		0x04	Received a new Extended Artist label for selection in location 0
ADF_ExtSongLabel_0		0x08	Received a new Extended Song label for selection in location 0
ADF_Text_0		0x10	Received new Text message for selection in location 0
ADF_ProgramStart_0		0x20	Received a new Program Start indication for selection in location 0
ADF_ProgramEnd_0		0x40	Received a new Program End indication for selection in location 0
ADF_ProgramId_0		0x80	Received a new Program ID indication for selection in location 0
ADF_Other_0	0x47	0x01	Received a new Other message for selection in location 0
BIC_ServiceLabel_1		0x02	Received a new Service label for selection in location 1
BIC_ProgramTypeLabel_1		0x04	Received a new Program type label for selection in location 1
BIC_ServiceSelection_1		0x08	Received a new Service Selection message for selection in location 1
BIC_SongArtistLabel_1		0x10	Received a new Song/Artist label for selection in location 1
BIC_ChannelReferenceLabel_1		0x20	Channel Reference Table changed for selection in location 1
ADF_ExtArtistLabel_1		0x40	Received a new Extended Artist label for selection in location 1
ADF_ExtSongLabel_1		0x80	Received a new Extended Song label for selection in location 1
ADF_Text_1	0x48	0x01	Received new Text message for selection in location 1
ADF_ProgramStart_1		0x02	Received a new Program Start indication for selection in location 1
ADF_ProgramEnd_1		0x04	Received a new Program End indication for selection in location 1
ADF_ProgramId_1		0x08	Received a new Program ID indication for selection in location 1
ADF_Other_1		0x10	Received a new Other message for selection in location 1
ChNumChanged_0		0x20	The channel number is changed for SID extracted into location 0
ChNumChanged_1		0x40	The channel number is changed for SID extracted into location 1

HOST_ERRINTEH - HOST_ERRINTEL

	7	6	5	4	3	2	1	0
4A	INTE[15:8]							
49	INTE[7:0]							

Address : 0x4A - 0x49
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
These registers are associated to error condition inside the STA450A.
The STA450A contains a 16 bit interrupt register associated with 16 bit enable register.
A bit set in this register will enable the generation of an external interrupt on the interrupt line. The interrupt associated with each bit is given in the register INT description.

HOST_ERRINTH - HOST_ERRINTL

	7	6	5	4	3	2	1	0
4C	INT[15:8]							
4B	INT[7;0]							

Address : 0x4B - 0x4C
Type : R/W
Software Reset : NA
Hardware Reset : NA
Description
These registers are associated to error condition inside the STA450A.
A bit set in this table indicates the corresponding event has been occurred. Whenever the corresponding bit in the INTE table has been set an external interrupt is generated.

To clear the bit a fast command has to be issued (see command description).

Name	Reg	Bit	Comment
Service Layer Lost Synchronization	0x4B	0x01	Service Layer Lost synchronisation due to communication error with CDEC
TSCC RSError		0x02	TSCC have RS errors
Bitstream Communication Failure		0x04	The format of communication is not compatible with the bitstream input I/F. The DSP need to be restarted
Service Layer incorrect status		0x08	Service Layer have reached an incorrect status.
NoBacFromCdec		0x10	The BAC wasn't received from the CDEC in the last frame
Rollback		0x20	Error from decryption
PicIOError		0x40	Communication error with the CAP device
Audio Decoder Not Working		0x80	Error in the audio decoder
NV Memory Unreadable	0x4C	0x01	Data corrupted in the NVM
ExtractionError_0		0x02	Some error occurs during extraction of channel in location 0, to obtain detailed situation System Controller must raise a command. At the time of this command is raised the DSP clear the error buffer
ExtractionError_1		0x04	Some error occurs during extraction of channel in location 1, to obtain detailed situation System Controller must raise a command. At the time of this command is raised the DSP clear the error buffer

HOST_STARTUP

	7	6	5	4	3	2	1	0
4D								

Address : 0x4D
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description

Writing 0x01 in this register enables the STA450A sw to leave the wait mode and to start normal process.
Using this register is also possible select special mode for silicon evaluation
Before to set this register is mandatory to program the system PLL , the Audio Fractional PLL and to configure the PCM output format according to the application DAC.

HOST_Pcmdiv

	7	6	5	4	3	2	1	0
4E								

Address : 0x4E
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description

This register is red by STA450A before to leave the wait mode.
The SCLK signal is derived from the clock PCMCLK.

HOST_pcmdiv = (0_FAC/64) -1 in 16 bit mode
HOST_pcmdiv = (0_FAC/128) -1 in 18/20/24 bit mode
If Pcm_div is set to 0, the SCLK frequency is equal to the PCMCLK frequency.

HOST_Pcmcnf

	7	6	5	4	3	2	1	0
4F								

Address : 0x4F
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description

This register is red by STA450A before to leave the wait mode.
Pcm_prec
Bit [1:0] 00: 16 bits mode.
01: 18 bits mode
10: 20 bits mode
11: 24 bits mode

Invert_sclk
Bit 2 0: LRCLK and PCM_OUT sampled on the falling edge of the SCLK
1: LRCLK and PCM_OUT sampled on the raising edge of the SCLK
Format
Bit 3 0: the output is in I2S format.
1: the output is in SONY format.
Invert_lrclk
Bit 4 0: LRCLK = 0 (low) will select the left channel.
1: LRCLK = 1 (high) will select the left channel
Pcm_dif
Bit 5 0: data are in the last SCLK cycles of LRCLK (right aligned)
1: data are in the first SCLK cycles of LRCLK (left aligned)
Pcm_ord
Bit 6 0: the transmission is done LSB first.
1: the transmission is done MSB first.
Pcm_iec_chansel
Bit 7 0: no iec958 output.
1: iec958 output, data on I^2S pin (PCSD) are no more valid.

HOST_APLL48_LSB

	7	6	5	4	3	2	1	0
50								

Address : 0x50
Type : R/W
Software Reset : NA
Hardware Reset : NA

HOST_APLL48_MSB

	7	6	5	4	3	2	1	0
51								

Address : 0x51
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
The HOST_APLL48_LSB and the HOST_APLL48_MSB are considered logically concatenated and contain the fractional values for the Audio Fractional PLL for the Fs = 48KHz family.
The registers have to be programmed before the start_up of the DSP.

HOST_APLL48_XDIV

	7	6	5	4	3	2	1	0
52								

Address : 0x52

Type : R/W
Software Reset : NA
Hardware Reset : NA

HOST_APLL48_MDIV

	7	6	5	4	3	2	1	0
53								

Address : 0x53
Type : R/W
Software Reset : NA
Hardware Reset : NA

HOST_APLL48_NDIV

	7	6	5	4	3	2	1	0
54								

Address : 0x54
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
The HOST_APLL48_XDIV, HOST_APLL48_MDIV and HOST_APLL48_NDIV registers are used to configure the X, M and N divider of the Audio Fractional PLL for the Fs = 48KHz family.
The registers have to be programmed before the start_up of the DSP.

.**HOST_APLL441_LSB**

	7	6	5	4	3	2	1	0
50								

Address : 0x55
Type : R/W
Software Reset : NA
Hardware Reset : NA

HOST_APLL441_MSB

	7	6	5	4	3	2	1	0
51								

Address : 0x56
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
The HOST_APLL441_LSB and the HOST_APLL441_MSB are considered logically concatenated and

contain the fractional values for the Audio Fractional PLL for the Fs = 44.1KHz family.
The registers have to be programmed before the start_up of the DSP.

HOST_APLL441_XDIV

	7	6	5	4	3	2	1	0
52								

Address : 0x57
Type : R/W
Software Reset : NA
Hardware Reset : NA

HOST_APLL441_MDIV

	7	6	5	4	3	2	1	0
53								

Address : 0x58
Type : R/W
Software Reset : NA
Hardware Reset : NA

HOST_APLL441_NDIV

	7	6	5	4	3	2	1	0
54								

Address : 0x59
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
The HOST_APLL441_XDIV, HOST_APLL441_MDIV and HOST_APLL441_NDIV registers are used to configure the X, M and N divider of the Audio Fractional PLL for the Fs = 44.1KHz family.
The registers have to be programmed before the start_up of the DSP

ENABLE_IT432

	7	6	5	4	3	2	1	0
5A								

Address : 0x5A
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Enabling of the Audio Fractional PLL control through the external 432ms interrupt on pin PLL_SYNC.
0: Disabled
1: Enabled

IT432_CONF

	7	6	5	4	3	2	1	0
5B								

Address : 0x5B
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Allows to configure if the 432ms interrupt is rising or falling edge sensitive.
0: sensitive to falling edge
1: sensitive to rising edge

HOST_MAXDEV

	7	6	5	4	3	2	1	0
5C								

Address : 0x5C
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Divider factor for deviation value window

HOST_Decoder BitRate

	7	6	5	4	3	2	1	0
5D								

Address : 0x5D
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Audio decoder bit rate / 1K

HOST_BitstreamSynchro

	7	6	5	4	3	2	1	0
5E								

Address : 0x5E
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
01: Not synchronised
10: Synchronised

HOST_DataPortBitRate

	7	6	5	4	3	2	1	0
5F								

Address : 0x5F
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Data port bit rate / 1K (for bitrates ≥ 128Kbps 0xFF is reported)
10: Synchronised

HOST_EVENTINTE 4-5

	7	6	5	4	3	2	1	0
66	INTE[15:8]							
67	INTE[7:0]							

Address : 0x66 - 0x67
Type : R - W
Software Reset : NA
Hardware Reset : NA

Description
These registers are associated to event condition inside the SDEC.
The SDEC contains a 16 bits interrupt register associated with 16 bits enable register.
A bit set in this register will enable the generation of an external interrupt on the interrupt line.
The interrupt associated with each bit is given in the register INT description.

HOST_EVENTINT 4-5

	7	6	5	4	3	2	1	0
68	INT[15:8]							
69	INT[7:0]							

Address : 0x68 - 0x69
Type : R - W
Software Reset : NA
Hardware Reset : NA

Description
These registers are associated to event condition inside the SDEC.
A bit set in this table indicates the corresponding event has been occurred. Whenever the corresponding bit in the INTE table has been set an external interrupt is generated.
To clear the bit a fast command has to be issued (see command description).

Name	Reg	Bit	Comment
Fast extraction Loc0	0x68	0x01	Channel fast extraction for Loc0 successful
Fast extraction Loc1		0x02	Channel fast extraction for Loc1 successful

Name	Reg	Bit	Comment
CRW monitor list change		0x04	Channel Reference Word list changed
Program Type (BIC 3) change		0x08	Program Type (BIC 3) changed
Artist and Song change		0x10	Artist and Song changed
Service Label (BIC 5) change		0x20	Service Label (BIC 5) changed
Program Label change		0x40	Program Label changed
BIC Extended Song label Loc0		0x80	Extended Song label for Loc0 changed
BIC Text Label Loc0	0x69	0x01	Text Label for Loc0 changed
BIC Extended Artist label Loc0		0x02	Extended Artist label for Loc0 changed
BIC Extended Song label Loc1		0x04	Extended Song label for Loc1 changed
BIC Extended Artist label Loc1		0x08	Extended Artist label for Loc1 changed
BIC Text Label Loc1		0x10	Text Label for Loc1 changed
RFU		0x20	
RFU		0x40	
RFU		0x80	

HOST_ AudioDecResult

	7	6	5	4	3	2	1	0
6B								

Address : 0x6B
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Upper nibble defines the decoder type:
0x10: AMBE
0x20: AAC
Lower nibble defines the Audio decoder result.

0x00	data ok
0x01	error concealment (only for AAC)
0x02	synchronization lost, output muted

HOST_ MFC

	7	6	5	4	3	2	1	0
6C								

Address : 0x6C
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Master Frame Counter LSB part (7 bits).

HOST_Page Ctrl

	7	6	5	4	3	2	1	0
7F	More	Byte Number-1						

Address : 0x7F
Type : R/W
Software Reset : NA
Hardware Reset : NA

Description
Bit 7 :more page to be download.
This bit is used also as Hand Shake bit.
The STA450A sets the bit as soon as the page is available and the System Controller clears the bit using the dedicated Fast command (see command description) as soon as the page has been read. In case the System Controller sends a new command and the MORE bit is set, the STA450A clears it automatically.
Bit [6:0] :number - 1 of valid bytes in the current page

3.0 DSP COMMANDS QUICK REFERENCE

References: Service Layer Specification
XM Stack API Specification

3.1 Categories

INIT 0x0X
SELECT 0x1X
INFO 0x2X
MISC 0x3X
FASTCMD 0x5X

3.2 COMMANDS

Command	OpCode	Category	Description
DSP_PWRUP_REQ	0x00	INIT	DSP Initialization command
DSP_PWRDWN_REQ	0x01	INIT	DSP Power Down command
FORCE_UPDATE	0x02	INIT	Force Update command
CHANGE_DAC_CONF	0x03	INIT	Change DSP DAC configuration command
HARD_MUTE	0x04	INIT	Hard mute/play command
FADE_IN_OUT	0x05	INIT	Fade In/Fade Out command
SOFT_MUTE	0x06	INIT	Soft mute/play command
DATAPORT_CH_FREQUENCY	0x07	INIT	Change the divider for Data port
SERVICE_CHECK	0x10	SELECT	Check the status of a service using SID

Command	OpCode	Category	Description
CHANNEL_CHECK	0x11	SELECT	Check the status of a service using channel number
CHANNEL_UP_DOWN	0x12	SELECT	Channel search command
PROGRAM_UP_DOWN	0x13	SELECT	Channel search trough Program type command
SERVICE_EXTRACT	0x14	SELECT	Service extraction command
EXTRACT_ERROR	0x15	SELECT	Request service extraction error information
SERVICE_CANCEL	0x16	SELECT	Cancel a service currently extracted command
SERVICE_ROUTING	0x17	SELECT	Change the routing of an extracted command
CHNCHECK_LIST_REQ	0X18	SELECT	
EXTRACT_STATUS_REQ	0x19	SELECT	After Fast extraction command the result is reported using this command
CHNSTATUS_LIST_REQ	0x1A	SELECT	
LABELMON_LIST_REQ	0x1B	SELECT	
SIB1_REQUEST	0x20	INFO	SIB 1 request
SERV_LABEL_REQ (SIB 2)	0x21	INFO	SIB 2 request
ARTSNG_LABEL_REQ (SIB 3)	0x22	INFO	SIB 3 request
PTY_LABEL_REQ	0x23	INFO	Program Type label request
ALL_LABEL_REQ	0x24	INFO	All label request
ADF_PROGRAM_ID_REQ	0x25	INFO	Program ID request
ADF_PROGRAM_START_REQ	0x26	INFO	Program Start request
ADF_PROGRAM_END_REQ	0x27	INFO	Program End request
ADF_EXTARTISTLBL_REQ	0x28	INFO	Extended Artist label request
ADF_EXTSONGLBL_REQ	0x29	INFO	Extended Song label request
ADF_TEXT_REQ	0x2A	INFO	Text message request
BIC_TEXT_REQ	0x2C	INFO	BIC Text request
CHANNEL_REF_REQ	0x2D	INFO	Channel Table request
ARTSNGCHG_LIST_REQ	0x2E	INFO	
PROGTYPE_LIST_REQ	0x2F	INFO	
HWID_READ_REQ	0x31	MISC	HW ID request
UTC_REQ	0x32	MISC	Universal Time Code and Master Frame Counter request
CLEAR_DATA_REQ	0x34	MISC	
PROGLABEL_LIST_REQ	0x36	MISC	
BIC_LABEL_REQ	0x37	MISC	Ram

Command	OpCode	Category	Description
DISPLAY_MASK_REQ	0x4C	SERVICE	This command must be issued before power up
SDEC_VER_REQ	0x4D	SERVICE	This command must be issued before power up The system controller can selected the version of the SDEC
FAST_CLEAR_EVENT	0x50	FASTR	Clear events and error bit according to applied mask
FAST_DISABLE_REQ	0x51	FASTR	Clear events and error interrupt enable bit according to applied mask
FAST_ENABLE _REQ	0x52	FASTR	Set events and errors interrupt enable bits according to applied mask
FAST_EXTRACT_REQ	0x53	FASTR	Service extract in fast mode

3.3 DSP COMMAND DESCRIPTION

The commands are composed by two parts: the REQ one (request) and the CFM one (confirmation).

The REQ has to be written in the Command registers (0x6D - 0x7E); the CFM has to be read in the Data Page registers (0x80 - 0xFF).

After a REQ has been issued the STA450A replies with a CFM within 5 MFP (5 x 432ms).

3.3.1 DSP_PWRUP_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	DSP_PWRUP_REQ	0x00	DSP Initialization Request header
1	serv_label_size	8, 12, 16	Display size for the service label.
2	gen_label_size	8, 12, 16	Display size for the song label, artist label, and program label.

BYTE	FIELD	VALUE	DESCRIPTION
0	DSP_ PWRUP _CFM	0x80	DSP Initialization Confirm header
1	error_code	0x00 – NO ERROR 0x01 – CAP_IO_ERROR 0x02 – INVALID_NVM_IMAGE	DSP power up error status
2	init_complete_status	0x04 – ACT_PRESENT 0x00 – ACT_ABSENT	Activation Status If both encryption keys are stored in memory, then DSP reports ACT_PRESENT. If no encryption keys are stored in memory, then DSP reports ACT_ABSENT.
3	last_ear	0 – 3	Last saved Entered Authorization Rating (EAR)
4	dsp_sw_version	0x00 – 0xFF	DSP software version
5 – 8	dsp_sw_date	Byte 5: month in BCD. Byte 6: day in BCD. Bytes 7,8: Year in BCD.	DSP release date. The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0

9 – 12	last_crw0		Last Channel Reference Word (CRW) for location zero Reference: Service Layer Specification
13	last_sc_type0		Last Service Component type for location zero Reference: Service Layer Specification
14 – 17	last_crw1		Last Channel Reference Word (CRW) for location one Reference: Service Layer Specification
18	last_sc_type1		Last Service Component type for location one. Reference: Service Layer Specification
19 – 26	Hwid		Hardware Identification

3.3.2 DSP_PWRDWN_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	DSP_PWRDWN_REQ	0x01	DSP Power Down Request header

BYTE	FIELD	VALUE	DESCRIPTION
0	DSP_PWRDWN_CFM	0x81	DSP Power Down Confirm header
1 – 2	error_code	0x01 – CAP IO Error 0x02 – Invalid image	Power down error code.

3.3.3 FORCE_UPDATE

BYTE	FIELD	VALUE	DESCRIPTION
0	FORCE_UPDATE_REQ	0x02	Force Update Request header

BYTE	FIELD	VALUE	DESCRIPTION
0	FORCE_UPDATE_CFM	0x82	Force Update Request header

3.3.4 CHANGE_DAC_CONF

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANGE_DAC_CONF_REQ	0x03	DSP DAC configuration Request header
1	Pcm_divider		
2	Pcm_configuration		

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANGE_DAC_CONF_CFM	0x83	DSP DAC configuration confirmation

3.3.5 HARD_MUTE

BYTE	FIELD	VALUE	DESCRIPTION
0	HARD_MUTE_REQ	0x04	DSP hard mute Request header
1	Pcm_playmue	1 - play 2 - hard mute	

BYTE	FIELD	VALUE	DESCRIPTION
0	HARD_MUTE_CFM	0x84	DSP hard mute confirmation

3.3.6 FADE_IN_OUT

BYTE	FIELD	VALUE	DESCRIPTION
0	FADE_IN_OUT _REQ	0x05	DSP Fade In/Out Request header
1	nr_fadeout_frames	0x03 – Default	Number of fadeout frames until mute state is reached for frequencies less than 32kbps
2	nr_fadein_frames	0x05 – Default	Number of fadein frames until regular state is reached for frequencies less than 32kbps
3	nr_valide_frames	0x05 – Default	Number of valid frames before fadein starts for frequencies less than 32kbps
4	cmftNoiseLimit	0x00 – Min 0x01 – Default 0x64 (100) – Max	This parameter give the maximum level for 'comfort noise'. This must be the percentage value respect to the maximum (for frequencies less than 32kbps)
5	nr_fadeout_frames	0x06 – Default	Number of fadeout frames until mute state is reached for frequencies greater than 32kbps
6	nr_fadein_frames	0x05 – Default	Number of fadein frames until regular state is reached for frequencies greater than 32kbps
7	nr_valide_frames	0x05 – Default	Number of valid frames before fadein starts for frequencies greater than 32kbps
8	cmftNoiseLimit	0x00 – Min 0x01 – Default 0x64 (100) – Max	This parameter give the maximum level for 'comfort noise'. This must be the percentage value respect to the maximum (for frequencies greater than 32kbps)

BYTE	FIELD	VALUE	DESCRIPTION
0	FADE_IN_OUT _CFM	0x85	DSP Fade In/Out confirmation
1	Command status		0x01 = success 0xFF = command not executed

3.3.7 SOFT_MUTE

BYTE	FIELD	VALUE	DESCRIPTION
0	SOFT_MUTE_REQ	0x06	DSP soft mute Request header
1	Soft_playmute	0 – play 1 – soft mute	

BYTE	FIELD	VALUE	DESCRIPTION
0	SOFT_MUTE_CFM	0x86	DSP soft mute confirmation

3.3.8 DATAPORT_CH_FREQUENCY

BYTE	FIELD	VALUE	DESCRIPTION
0	DP_CF_REQ	0x07	Data port change frequency Request header
1	Data port frequency divider	Max. 0x05 – 12 Mhz Default 0x1E – 2 Mhz Min. 0x3C – 1Mhz	System clock divider. E.g. Divider = 0x1E (30) DpClk = (SysClk / 12); DpClk = (60 Mhz / 12) = 2 Mhz

BYTE	FIELD	VALUE	DESCRIPTION
0	DP_CF_CFM	0x87	Data port change frequency confirmation

3.3.9 SERVICE_CHECK

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_CHECK_REQ	0x10	Service Check Request header
1	Sid	1-255	Service identifier
2	extract_type	0x00 – Audio 0x01 – Data	Service type to extract. MMI can extract the audio or data service component.

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_CHECK_CFM	0x90	Service Check Confirm header
1	service_check_status	**bit 7 – Service Present** 0x80 – SID_PRESENT 0x00 – SID_ABSENT **bit 6 – PC Forma**t 0x40 – PC_STANDARD 0x00 – PC_NONSTANDARD **bit 5 – Service On-Air** 0x20 – SERVICE_ONAIR 0x00 – SERVICE_OFFAIR **bit 4 – Service Type** 0x10 – TYPE_PRESENT 0x00 – TYPE_ABSENT **bit 3 – Service Free** 0x08 – SERVICE_FREE 0x00 – SERVICE_NOTFREE **bit 2 – Activation** 0x04 – ACT_PRESENT 0x00 – ACT_ABSENT **bit 1 – Authorization** 0x02 – AUTH_PRESENT 0x00 – AUTH_ABSENT **b0 – Reserved**	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0
2	channel_number	1 – 255	channel number for the service
3,4,5,6	crw		Channel Reference Word Reference: Service Layer Specification.

3.3.10 CHANNEL_CHECK

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_CHECK_REQ	0x11	Channel Check Request header
1	channel_number	1-255	Channel number
2	extract_type	0x00 – Audio 0x01 – Data	Service component type to extract. MMI can extract the audio or data service component.

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_CHECK_CFM	0x91	Channel Check Confirm header
1	service_check_status	**bit 7 – Service Present** 0x80 – SID_PRESENT 0x00 – SID_ABSENT **bit 6 – PC Format** 0x40 – PC_STANDARD 0x00 – PC_NONSTANDARD **bit 5 – Service On-Air** 0x20 – SERVICE_ONAIR 0x00 – SERVICE_OFFAIR **bit 4 – Service Type** 0x10 – TYPE_PRESENT 0x00 – TYPE_ABSENT **bit 3 – Service Free** 0x08 – SERVICE_FREE 0x00 – SERVICE_NOTFREE **bit 2 – Activation** 0x04 – ACT_PRESENT 0x00 – ACT_ABSENT **bit 1 – Authorization** 0x02 – AUTH_PRESENT 0x00 – AUTH_ABSENT **b0 – Reserved**	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0
2,3,4,5	crw		Channel Reference Word Reference: Service Layer Specification
6	sib1_status	0x00 – SIB1_ABSENT 0x01 – SIB1_PRESENT	If service information block 1 was not received, then DSP returns SIB1_ABSENT. If service information block 1 was received, then DSP returns SIB1_PRESENT
7 to 14	service_block1		Service Information Block 1 Reference: Service Layer Specification.

3.3.11 CHANNEL_UP_DOWN

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_UPDWN_REQ	0x12	Channel Up/Down Request header
1	Direction	0x00 – UP 0x01 - DOWN	Direction to search
2	channel_start_number	1-255	Channel start number
3	channel_end_number	1-255	Channel end number
4	extract_type	0x00 – Audio 0x01 – Data	Service type to extract. MMI can extract the audio or data service component.

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_UPDWN_CFM	0x92	Channel Up/Down Confirm header
1	search_status	**bit 7 – Search Outcome** 0x80 – SEARCH_SUCCESSFUL 0x00 – SEARCH_FAIL **bit 6 – Extract Type** 0x40 – TYPE_PRESENT 0x00 – TYPE_ABSENT **bit 5 – Reserved** **bit 4 – Reserved** **bit 3 – Reserved** **bit 2 – Reserved** **bit 1 – Reserved** **bit 0 – Reserved**	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0
2	channel_number	1 – 255	channel number
3,4,5,6	crw		Channel Reference Word Reference: Service Layer Specification
7	sib1_status	0x00 – SIB1_ABSENT 0x01 – SIB1_PRESENT	If service information block 1 was not received, then DSP returns SIB1_ABSENT. If service information block 1 was received, then DSP returns SIB1_PRESENT.
8 to 15	service_block1		Service Information Block 1 Reference: Service Layer Specification

3.3.12 PROGRAM_UP_DOWN

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_UPDWN_REQ	0x13	Program Up/Down Request header
1	Direction	0x00 – UP 0x01 - DOWN	Direction to search
2	channel_start_number	1-255	Channel start number
3	channel_end_number	1-255	Channel end number
4	program_type	1-31	Program type
5	extract_type	0x00 – Audio 0x01 – Data	Service type to extract. MMI can extract the audio or data service component.

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_UPDWN_CFM	0x93	Program Up/Down Confirm header
1	search_status	**bit 7 – Search Outcome** 0x80 – SEARCH_SUCCESSFUL 0x00 – SEARCH_FAIL **bit 6 – Extract Type** 0x40 – TYPE_PRESENT 0x00 – TYPE_ABSENT **bit 5 – Reserved** 0x20 – PROGRAM_PRESENT 0x00 – PROGRAM_ABSENT **bit 4 – Reserved** **bit 3 – Reserved** **bit 2 – Reserved** **bit 1 – Reserved** **bit 0 – Reserved**	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0
2	channel_number	1 – 255	channel number for the service

3,4,5,6	crw		Channel Reference Word Reference: Service Layer Specification
7	sib1_status	0x00 – SIB1_ABSENT 0x01 – SIB1_PRESENT	If service information block 1 was not received, then DSP returns SIB1_ABSENT. If service information block 1 was received, then DSP returns SIB1_PRESENT.
8 to 15	service_block1		Service Information Block 1 Reference: Service Layer Specification

3.3.13 SERVICE_EXTRACT

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_EXTRACT_REQ	0x14	Service Extract Request header
1	location	0x00 – location 0 0x01 – location 1	Since DSP can extract two services simultaneously, XM stack assigns an extraction location.
2 – 3	pc_detect_time	0x0001 to 0xFFFF	Number of master frame that the DSP waits for the primary PCID or secondary PCID.
4	sid	1 – 255	Service Identifier
5	sc_type	0x03 – Audio AMBE 0x05 – Audio AAC+ 0x0A – Trans. Data	Service Component Type Reference: Service Layer Specification; page 21
6	primary_pcid	5 – 255	Primary Payload Channel Identification Reference: Transport Layer Specification
7	Secondary_pcid	5 – 255	Secondary Payload Channel Identification Reference: Transport Layer Specification
8	Routing	0x00 – No routing 0x01 – AUD_PORT 0x02 – DAT_PORT 0x03 – BOTH_PORT	Routing information A data service component can only be routed to the data port (DAT_PORT). An audio service component be routed to audio port (AUD_PORT), data port (DAT_PORT) or both the audio and data port (BOTH_PORT).

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_EXTRACT_CFM	0x94	Service Extract Confirm header
1	Location	0x00 – location 0 0x01 – location 1 0xFF – command refused	Location assigned to the service. If the command was refused the location is set to 0xFF. This can happen if the extraction of a component with routing for audio port is commanded and a component with routing audio port is already extracted in the other location or if an inexistent location is passed to the DSP

2	extract_error_status	**bit 7 – PC Detect** 0x80 – PCID_UNDETECTED 0x00 – PCID_DETECTED **bit 6 – Service Detect** 0x40 – SID_UNDETECTED 0x00 – SID_DETECTED **bit 5 – SC Type Detect** 0x20 – SCT_UNDETECTED 0x00 – SCT_DETECTED **bit 4 – Authorisation** 0x10 – AUTH_MISSING 0x00 – AUTH_OK **bit 3 – EAR Low** 0x08 – EAR_LOW 0x00 – EAR_OK **bit 2 – RAT Block** 0x04 – RAT_BLOCK 0x00 – RAT_OK **bit 1 – RAT Low** 0x02 – RAT_LOW 0x00 – RAT_OK **b0 – TOD Rollback** 0x01 – TOD_ROLLBACK 0x00 – TOD_OK	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0

3.3.14 EXTRACT_ERROR

BYTE	FIELD	VALUE	DESCRIPTION
0	EXTRACT_ERROR_REQ	0x15	Extract Error Request header
1	Location	0x00 – location 0 0x01 – location 1 0x02 – generic error (TEST only)	DSP reports errors for the selected location or errors related to the data analysis and extraction (test only).

BYTE	FIELD	VALUE	DESCRIPTION
0	EXTRACT_ERROR_CFM	0x95	Extract Error Confirm header
1	extract_error_status	**bit 7 – PC Detect** 0x80 – PCID_UNDETECTED 0x00 – PCID_DETECTED **bit 6 – Service Detect** 0x40 – SID_UNDETECTED 0x00 – SID_DETECTED **bit 5 – SC Type Detect** 0x20 – SCT_UNDETECTED 0x00 – SCT_DETECTED **bit 4 – Authorisation** 0x10 – AUTH_MISSING 0x00 – AUTH_OK **bit 3 – EAR Low** 0x08 – EAR_LOW 0x00 – EAR_OK **bit 2 – RAT Block** 0x04 – RAT_BLOCK 0x00 – RAT_OK **bit 1 – RAT Low** 0x02 – RAT_LOW 0x00 – RAT_OK **b0 – TOD Rollback** 0x01 – TOD_ROLLBACK 0x00 – TOD_OK	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0. If the command is refused 0xFF is returned

3.3.15 SERVICE_CANCEL

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_CANCEL_REQ	0x16	Service Cancel Request header
1	location	0x00 – location 0 0x01 – location 1	Since DSP can extract two services simultaneously, XM stack specifies which location to cancel.

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_CANCEL_CFM	0x96	Service Cancel Confirm header
1	location	0x00 – location 0 0x01 – location 1	DSP reports the extraction location that was cancelled. If the command is refused 0xFF is returned

3.3.16 SERVICE_ROUTING

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_ROUTING_REQ	0x17	Service Routing Request header
1	location	0x00 – location 0 0x01 – location 1	Since DSP can extract two services simultaneously, XM stack specifies which location to change the routing
2	routing	0x00 – No routing 0x01 – AUDIO_PORT 0x02 – DATA_PORT 0x03 – BOTH_PORT	Routing information

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_ROUTING_CFM	0x97	Service Routing Confirm header
1	location	0x00 – location 0 0x01 – location 1	Location that changed routing. If the command is refused 0xFF is returned
2	routing	0x00 – No routing 0x01 – AUDIO_PORT 0x02 – DATA_PORT 0x03 – BOTH_PORT	New routing configuration. If the command is refused the old routing status is returned

3.3.17 CHCHECK_LIST_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	CHNCHECK_LIST_REQ	0x18	Channel Check List Request Header
1	extract_type	0x00 – Audio 0x01 - Data	MMI can extract the audio or data service component.
2	prog_type_flag	0x00 – FALSE 0x01 - TRUE	If the MMI is requesting only the channels belonging to a program type, then the prog_type_flag is set to TRUE. If the MMI is requesting all channels, then the prog_type_flag is set to FALSE.
3	program_type	1 - 31	If the prog_type_flag is set to TRUE, then the SDEC reports the service labels that only belongs to this program type.

BYTE	FIELD	VALUE	DESCRIPTION
0	CHNCHECK_LIST_CFM	0x98	Channel Check List Confirm Header
1	Command_status	0x01 - succesful 0xFF - failure	
2	channel_bit_mask00	bit 7 – channel 007 bit 6 – channel 006 bit 5 – channel 005 bit 4 – channel 004 bit 3 – channel 003 bit 2 – channel 002 bit 1 – channel 001 bit 0 – channel 000	Channel mask 0 – channel status indication bits for channels 0 to channel 7.
3	channel_bit_mask01	bit 7 – channel 015 bit 6 – channel 014 bit 5 – channel 013 bit 4 – channel 012 bit 3 – channel 011 bit 2 – channel 010 bit 1 – channel 009 bit 0 – channel 008	Channel mask 1 – channel status indication bits for channels 8 to channel 15.
4	channel_bit_mask02	bit 7 – channel 023 bit 6 – channel 022 bit 5 – channel 021 bit 4 – channel 020 bit 3 – channel 019 bit 2 – channel 018 bit 1 – channel 017 bit 0 – channel 016	Channel mask 3 – channel status indication bits for channels 16 to channel 23.
……	……	……	…….
……	……	……	…….
……	……	……	…….
33	channel_bit_mask31	bit 7 – channel 255 bit 6 – channel 254 bit 5 – channel 253 bit 4 – channel 252 bit 3 – channel 251 bit 2 – channel 250 bit 1 – channel 249 bit 0 – channel 248	Channel mask 32 – channel status indication bits for channels 248 to channel 255.

3.3.18 EXTRACT_STATUS_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	EXTRACT_STATUS_REQ	0x19	Extract Status Request header
1	location	0x00 – location 0 0x01 – location 1	Since SDEC can extract two services simultaneously, XM stack specifies which location to get the extraction errors

BYTE	FIELD	VALUE	DESCRIPTION
0	EXTRACT_STATUS_CFM	0x99	Extract Status Confirm header
1	location	0x00 – location 0 0x01 – location 1	Since SDEC can extract two services simultaneously, system controller assigns an extraction location.
2	Channel number	1-255	Extracted channel number
3	SID	1-255	SID of location 0 extraction.
4	Extraction status	0x00 – Not initiated 0x01 – In progress 0x02 - Ok 0x03 – Incorrect parameter in command.	
5	extract_error_status	**bit 7 – PC Detect** 0x80 – PCID_UNDETECTED 0x00 – PCID_DETECTED **bit 6 – Service Detect** 0x40 – SID_UNDETECTED 0x00 – SID_DETECTED **bit 5 – SC Type Detect** 0x20 – SCT_UNDETECTED 0x00 – SCT_DETECTED bit 4 – Authorization 0x10 – AUTH_MISSING 0x00 – AUTH_OK **bit 3 – Reserved** **bit 2 – RAT Block** 0x04 – RAT_BLOCKED 0x00 – RAT_UNBLOCKED **bit 1 – RAT Low** 0x02 – RAT_LOW 0x00 – RAT_OK **b0 – TOD Rollback** 0x01 – TOD_ROLLBACK 0x00 – TOD_OK	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0
6 – 8	Counter	Byte 6: LSB byte Byte 7: Middle byte Byte 8: MSB byte	If status=0, counter = 0 For all other status, counter is #x432ms for actual time, counter will be running free until a channel is selected or initialized on the next fast extract command

3.3.19 CHANNEL STATUS_LIST_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	CHNSTATUS_LIST_REQ	0x1A	Channel Status List Request Header
1	start_channel_number	1 - 255	The channel number where the SDEC starts the status check.
2	channel_number	1 to 32 max	number of selectable channels will be returned
3	extract_type	0x00- Audio 0x01 - Data	MMI can extract the audio or data service component.

4	prog_type_flag	0x00 – FALSE 0x01 - TRUE	If the MMI is requesting only the service labels belonging to a program type, then the prog_type_flag is set to TRUE. If the MMI is requesting all service labels, then the progtype_filter_flag is set to FALSE.
5	program_type	1 - 31	If the prog_type_flag is set to TRUE, then the SDEC reports the service labels that only belongs to this program type.
6	crw_report _flag	0x00 - FALSE 0x01 - TRUE	if the crw_report-flag is set to TRUE, then the SDEC reports the CRW for each selectable channel requested.
7	sib1_report_flag	0x00 – FALSE 0x01 - TRUE	if the sib1_report_flag is set to TRUE, then the SDEC reports the program types for each selectable channel requested.
8	sib2_report_flag	0x00 – FALSE 0x01 - TRUE	if the sib2_report_flag is set to TRUE, then the SDEC reports SIB2t (service selectable) for each selectable channel requested.
9	sib3_report_flag	0x00 – FALSE 0x01 - TRUE	if the sib3_report_flag is set to TRUE, then the SDEC reports SIB3(artist name and song title) for each selectable channel requested.

BYTE	FIELD	VALUE	DESCRIPTION
0	CHNSTATUS_LIST_CFM	0x9A	Channel Status List Confirm Header
1 - 2	size	0x0000 to 0x07A0 (bytes)	Size in byte of the Channel Status List Confirm response. The size excluding byte 0, 1, 2 MSB in byte 1. LSB in byte 2.
3	channel_number	1 - 255	The channel number of the first available service label.
4	service_check_status	**bit 7 – Reserved** **bit 6 - PC Format** 0x40 – PC_STANDARD 0x00 – PC_NONSTANDARD **bit 5 – Service On-Air** 0x20 – SERVICE_ONAIR 0x00 – SERVICE_OFFAIR **bit 4 – Reserved** **bit 3 – Service Free** 0x08 – SERVICE_FREE 0x00 – SERVICE_NOTFREE **bit 2 – Activation** 0x04 – ACT_PRESENT 0x00 – ACT_ABSENT **bit 1 – Authorization** 0x02 – AUTH_PRESENT 0x00 – AUTH_ABSENT **b0 - Reserved**	The bits are ordered from left to right. The MSB is on the left and the LSB is on the right. b7, b6, b5, b4, b3, b2, b1, b0

5 - 8	crw		Channel Reference Word Reference: Service Layer Specification Revison 1.2, page 38 and 56.
9	sib1_status	0x00 - SIB1_ABSENT 0x01 - SIB1_PRESENT	
10	program_type0	1 - 31	The channel's first program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type.
11	program_type1	1 - 31	The channel's second program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type.
12	program_type2	1 - 31	The channel's third program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type.
13	program_type3	1 - 31	The channel's fourth program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type
14	sib2_status	0x00 – SIB2_ABSENT 0x01 – SIB2_PRESENT	SIB2 status
15 to 30	service_label	ISO-8859-1 character set	The service label (SIB2). Note If sib2 is ABSENT, SDEC inserts 0x00 for all 16 characters.
31	sib3_status	0x00 – SIB3_ABSENT 0x01 – SIB3_PRESENT	SIB3 status
32 to 47	artist_name	ISO-8859-1 character set	artist_name Note If sib3 is ABSENT, SDEC inserts 0x00 for all 16 characters.
48 to 63	song_title	ISO-8859-1 character set	song title Note If sib3 is ABSENT, SDEC inserts 0x00 for all 16 characters.
……	……	……	……
……	……	……	……
……	……	……	……
size - 61	channel_number	1- 255	The channel number of the last available service label
size - 60	service_check_status	same as above	same as above
size - 59 to size -54	crw		Channel Reference Word Reference: Service Layer Specification Revison 1.2, page 38 and 56.
size - 55	sib1_status	0x00 – SIB1_ABSENT 0x01 – SIB1_PRESENT	sib1 status

size - 54	program_type0	1 - 31	The channel's first program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type.
size - 53	program_type1	1 - 31	The channel's second program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type.
size - 52	program_type2	1 - 31	The channel's third program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type.
size - 51	program_type3	1 - 31	The channel's fourth program type. Note if SIB1 is absent, SDEC inserts 0x00 for program type.
size - 50	sib2_satus	0x00 – SIB2_ABSENT 0x01 – SIB2_PRESENT	SIB2 status
size - 49 to size - 32	service_label	ISO-8859-1 character set.	The service label (SIB2) Note If sib2 is ABSENT, SDEC inserts 0x00 for all 16 characters
size - 33 to size - 31	sib3_satus	0x00 – SIB3_ABSENT 0x01 – SIB3_PRESENT	SIB3 status
size - 32 to size - 15	artist_name	ISO-8859-1 character set.	artist name Note If sib3 is ABSENT, SDEC inserts 0x00 for all 16 characters
size - 16 to size - 1	song_title	ISO-8859-1 character set.	song title. Note If sib3 is ABSENT, SDEC inserts 0x00 for all 16 characters

3.3.20 LABELMON_LIST_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	LABELMON_LIST_REQ	0x1B	Label Monitor List Request Header
1	monitor_request	0x00 – monitor label changes for all channels. 0x01 (NEW) 0x03 (ADD) – monitor label changes for channels listed below (from byte 23 to byte 16). 0x01 means the start of a new list. 0x03 means the updating of a list. In this way giving this command more than 1 time you can choose every channel number to test 0x02 (NEW) – 0x04 (ADD) monitor label changes for channels that belongs to program types listed below (from byte 2 to byte 16)	The channel monitor type. Note if monitor request set to 0x00 (all channels), this command is only two byte.

2	channel_program0	For channel number values is: 1 – 255 For program type values is : 1 – 31	If monitor request is set to 0x01, this byte is a channel number (1-255). If the monitor request is set to 0x02, this byte is a program type (1-31).
…..	…..	…..	…..
…..	…..	…..	…..
17	channel_program15	For channel number values is: 1 – 255 For program type values is: 1 – 31	If monitor request is set to 0x01, this byte is a channel number (1-255). If the monitor request is set to 0x02, this byte is a program type (1-31).

BYTE	FIELD	VALUE	DESCRIPTION
0	LABELMON_LIST_CFM	0x9B	Label Monitor List Confirm Header
1	monitor_request	0x00 – monitor label changes for all channels. 0x01/0x03 – monitor label changes for channels listed below (from byte 23 to byte16) 0x02/0x04 – monitor label changes for channels that belongs to program types listed below (from byte 2 to byte 16)	The channel monitor type. Note if monitor request set to 0x00 (all channels), this command is only two byte.

3.3.21 SIB1_REQUEST

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_SELECTION_REQ	0x20	Service Selection Request header
1	channel_number	1-255	Channel number for which DSP returns the Service Selection contained in Service Information Block 1.

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_SELECTION_CFM	0xA0	Service Selection Confirm header
1	sib1_status	0x00 – SIB1_ABSENT 0x01 – SIB1_PRESENT	If SIB1 was not received, then DSP returns SIB1_ABSENT. If SIB2 was received, then DSP returns SIB1_PRESENT.
2	sid	1-255	Service identifier
3 – 10	service_block1		Service Information Block 1 (SIB1). The service selection is contained in SIB1. Reference: Service Layer Specification.

3.3.22 SERV_LABEL_REQ (SIB 2)

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_LABEL_REQ	0x21	Service Label Request header
1	channel_number	1-255	Channel number for which DSP returns the Service Label.

BYTE	FIELD	VALUE	DESCRIPTION
0	SERVICE_LABEL_CFM	0xA1	Service Label Confirm header
1	sib2_status	0x00 – SIB2_ABSENT 0x01 – SIB2_PRESENT	If SIB2 was not received, then DSP returns SIB2_ABSENT. If SIB2 was received, then DSP returns SIB2_PRESENT.
2	sid	1-255	Service identifier
3 – 18	service_block2		Service Information Block 2 (SIB2). The artist and song label are contained in SIB2. Reference: Service Layer Specification

3.3.23 ARTSNG_LABEL_REQ (SIB 3)

BYTE	FIELD	VALUE	DESCRIPTION
0	ARTSNG_LABEL_REQ	0x22	Artist/Song Label Request header
1	channel_number	1-255	Channel number for which DSP returns the Artist/Song Label.

BYTE	FIELD	VALUE	DESCRIPTION
0	ARTSNG_LABEL_CFM	0xA2	Artist/Song Label Confirm header
1	sib3_status	0x00 – SIB3_ABSENT 0x01 – SIB3_PRESENT	If SIB3 was not received, then DSP returns SIB3_ABSENT. If SIB3 was received, then DSP returns SIB3_PRESENT
2	sid	1-255	Service identifier
3 – 39	service_block3		Service Information Block 3 (SIB3). The artist and song label are contained in SIB3. Reference: Service Layer Specification.

3.3.24 PTY_LABEL_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_LABEL_REQ	0x23	Program Label Request header
1	program_type	1 – 31	Program type. The program type is used as an index into the program label table to obtain the program label.

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_LABEL_CFM	0xA3	Program Label Confirm header
1	program_label_status	0x00 – PROGLABEL_ABSENT 0x01 – PROGLABEL_PRESENT	If the program label was not received, then DSP returns PROGLABEL_ABSENT. If the program label was received, then DSP returns PROGLABEL_PRESENT.

2 – 17	program_label		Program label. Reference: Service Layer Specification

3.3.25 ALL_LABEL_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	ALL_LABEL_REQ	0x24	All Label Request header
1	channel_number	1-255	Channel number for which DSP returns the labels.
2	program_type	1 – 31	Program type The program type is used as an index into the program label table to obtain the program label.

BYTE	FIELD	VALUE	DESCRIPTION
0	ALL_LABEL_CFM	0xA4	All Label Confirm header
1	sib2_status	0x00 – SIB2_ABSENT 0x01 – SIB2_PRESENT	If the service information block 2 was not received, then DSP returns SIB2_ABSENT. If the service information block 2 was received, then DSP returns SIB2_PRESENT.
2 – 17	service_block2		Service Information Block 2 (SIB2). The service label is contained in SIB2. Reference: Service Layer Specification
18	sib3_status	0x00 – SIB3_ABSENT 0x01 – SIB3_PRESENT	If the service information block 3 was not received, then DSP returns SIB3_ABSENT. If the service information block 3 was received, then DSP returns SIB3_PRESENT.
19 - 55	service_block3		Service Information Block 3 (SIB3). The artist label and song label is contained in SIB3. Reference: Service Layer Specification
56	program_label_status	0x00 – PROGLABEL_ABSENT 0x01 – PROGLABEL_PRESENT	If the program label was not received, then DSP returns PROGLABEL_ABSENT. If the program label was received, then DSP returns PROGLABEL_PRESENT.
57 – 72	program_label		Program label. Reference: Service Layer Specification.

3.3.26 ADF_PROGRAM_ID_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_ID_REQ	0x25	Program Identification Request header
1	sid	1-255	Service identifier for which DSP returns the program Identification.

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_ID_CFM	0xA5	Program Identification Confirm header

1	adf_status	0x00 – ADF_ABSENT 0x01 – ADF_PRESENT	If the program Identification ADF was not received, then DSP returns ADF_ABSENT. If the program Identification ADF was received, then DSP returns ADF_PRESENT.
2 – 6	adf_progid		Program Identification ADF. Reference: Service Layer Specification.

3.3.27 ADF_PROGRAM_START_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_START_REQ	0x26	Program Start Request header
1	sid	1-255	Service identifier for which DSP returns the program start time.

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_START_CFM	0xA6	Program Start Confirm header
1	adf_status	0x00 – ADF_ABSENT 0x01 – ADF_PRESENT	If the program start ADF was not received, then DSP returns ADF_ABSENT. If the program start ADF was received, then DSP returns ADF_PRESENT.
2 – 3	adf_progst		Program Start ADF. Reference: Service Layer Specification.

3.3.28 ADF_PROGRAM_END_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_END_REQ	0x27	Program End Request header
1	sid	1-255	Service identifier for which DSP returns the program and time.

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGRAM_END_CFM	0xA7	Program End Confirm header
1	adf_status	0x00 – ADF_ABSENT 0x01 – ADF_PRESENT	If the program end ADF was not received, then DSP returns ADF_ABSENT. If the program end ADF was received, then DSP returns ADF_PRESENT.
2 – 3	adf_progend		Program End ADF. Reference: Service Layer Specification.

3.3.29 ADF_EXTARTISTLBL_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	EXTART_LABEL_REQ	0x28	Extended Artist Label Request header
1	sid	1-255	Service identifier for which DSP returns the extended artist label.

BYTE	FIELD	VALUE	DESCRIPTION

0	EXTART_LABEL_CFM	0xA8	Extended Artist Label Confirm header
1	adf_status	0x00 – ADF_ABSENT 0x01 – ADF_PRESENT	If extended artist ADF was not received, then DSP returns ADF_ABSENT. If extended artist ADF was received, then DSP returns ADF_PRESENT.
2 – 34	adf_extart		Extended Artist ADF. Reference: Service Layer Specification.

3.3.30 ADF_EXTSONGLBL_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	EXTSNG_LABEL_REQ	0x29	Extended Song Label Request header
1	sid	1-255	Service identifier for which DSP returns the extended song label.

BYTE	FIELD	VALUE	DESCRIPTION
0	EXTSNG_LABEL_CFM	0xA9	Extended Song Label Confirm header
1	adf_status	0x00 – ADF_ABSENT 0x01 – ADF_PRESENT	If extended song ADF was not received, then DSP returns ADF_ABSENT. If extended song ADF was received, then DSP returns ADF_PRESENT.
2 – 34	adf_extsng		Extended Song Label ADF. Reference: Service Layer Specification.

3.3.31 ADF_TEXT_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_TEXT_REQ	0x2A	Channel Text Request header
1	sid	1-255	Service identifier for which DSP returns the text.

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_TEXT_CFM	0xAA	Channel Text Confirm header
1	adf_status	0x00 – ADF_ABSENT 0x01 – ADF_PRESENT	If text ADF was not received, then DSP returns ADF_ABSENT. If text ADF was received, then DSP returns ADF_PRESENT.
2 – 260	adf_text		Text ADF. Reference: Service Layer Specification.

3.3.32 BIC_TEXT_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	BIC_TEXT_REQ	0x2C	BIC Text Request header
1	sid	1-255	Service identifier for which DSP returns the text.

BYTE	FIELD	VALUE	DESCRIPTION

0	BIC_TEXT_CFM	0xAC	BIC text Confirm header
1	BIC_text_status	0x00 –ABSENT 0x01 – PRESENT	If BIC text was not received, then DSP returns ABSENT. If BIC text was received, then DSP returns PRESENT.
2	sid	1-255	SID
3	scroll_ narrowcast	see specification	Scrolling and narrowcast information. Reference: Service Layer Specification; revision 1.4; section 8.11.
4	text_size	1-253	Length of the text message.
5 to text_size + 4	text	Extended ASCII	BIC Text Reference: Service Layer Specification; revision 1.4; section 8.11.

3.3.33 CHANNEL_REF_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_TABLE_REQ	0x2D	Channel Table Request header.
1	start_channel_number	1 - 225	DSP returns the channel reference table (CRT) segment from start_channel_number to start_channel_number + 30.

BYTE	FIELD	VALUE	DESCRIPTION
0	CHANNEL_TABLE_CFM	0xAD	Channel Table Confirm header
1	channel_number	1 - 225	Starting channel number.
2 – 255	channel_ref_table		Channel Reference Table Reference: Service Layer Specification.

3.3.34 ARTSNGCHG_LIST_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	ARTSNGCHG_LIST_REQ	0x2E	Artist/Song Change List Request Header

BYTE	FIELD	VALUE	DESCRIPTION
0	ARTSNG_LIST_CFM	0xAE	Artist/Song List Confirm Header
1	channel_bit_mask00	bit 7 – channel 007 bit 6 – channel 006 bit 5 – channel 005 bit 4 – channel 004 bit 3 – channel 003 bit 2 – channel 002 bit 1 – channel 001 bit 0 – channel 000	Channel mask 0 – artist/song status indication for channels from 0 to 7.

2	channel_bit_mask01	bit 7 – channel 015 bit 6 – channel 014 bit 5 – channel 013 bit 4 – channel 012 bit 3 – channel 011 bit 2 – channel 010 bit 1 – channel 009 bit 0 – channel 008	Channel mask 1 – artist/song status indication for channels from 8 to 15.
……	……	……	…….
……	……	……	…….
……	……	……	…….
32	channel_bit_mask31	bit 7 – channel 255 bit 6 – channel 254 bit 5 – channel 253 bit 4 – channel 252 bit 3 – channel 251 bit 2 – channel 250 bit 1 – channel 249 bit 0 – channel 248	Channel mask 1 – artist/song status indication for channels from 248 to 255.

3.3.35 PROGTYPE_LIST_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGTYPE_LIST_REQ	0x2F	Program Type List Request
1	REQ TYPE	0x01 – program types availables on lineup 0x02 – program types selectable by the user	

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGTYPE _LIST_CFM	0xAF	Program Type List Confirm header
1	CMD_RESPONSE	0x01 - command succesfull 0xFF – command fails	The command can fails in case that the command byte 1 is wrong (a non implemented number)
1	progtype_bit_mask00	bit 7 – program type 07 bit 6 – program type 06 bit 5 – program type 05 bit 4 – program type 04 bit 3 – program type 03 bit 2 – program type 02 bit 1 – program type 01 bit 0 – program type 00	Program type mask 0 – program type status indication bits for program type 0 to program type 7.
2	progtype_bit_mask01	bit 7 – program type 23 bit 6 – program type 22 bit 5 – program type 21 bit 4 – program type 20 bit 3 – program type 19 bit 2 – program type 18 bit 1 – program type 17 bit 0 – program type 16	Program type mask 1 – program type status indication bits for program type 8 to program type 15.

3	progtype_bit_mask02	bit 7 – program type 15 bit 6 – program type 14 bit 5 – program type 13 bit 4 – program type 12 bit 3 – program type 11 bit 2 – program type 10 bit 1 – program type 09 bit 0 – program type 08	Program type mask 2 – program type status indication bits for program type 16 to program type 23.
4	progtype_bit_mask03	bit 7 – program type 31 bit 6 – program type 30 bit 5 – program type 29 bit 4 – program type 28 bit 3 – program type 27 bit 2 – program type 26 bit 1 – program type 25 bit 0 – program type 24	Program type mask 3 – program type status indication bits for program type 24 to program type 31.

3.3.36 HWID_READ_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	HWID_READ_REQ	0x31	HWID Read Request header

BYTE	FIELD	VALUE	DESCRIPTION
0	HWID_READ_CFM	0xB1	HWID Read Confirm header
1 – 8	Hwid	Alphanumeric characters	Hardware Identification (HWID).

3.3.37 UTC_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	UTDC_REQ	0x32	HWID Read Confirm header

BYTE	FIELD	VALUE	DESCRIPTION
0	UNIVERSAL_TIME_CFM	0xB2	Universal Time Confirm header
1	utc_status	0x00 – UTC_ABSENT 0x01 – UTC_PRESENT	If UTC was not received, then DSP returns UTC_ABSENT. If UTC was received, then DSP returns UTC_PRESENT.
2 – 8	UTC		Universal Time. Reference: Transport Layer.
9 – 11	MFC		Master Frame Counter Reference: Transport Layer Specification.

3.3.38 CLEAR_DATA_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	CLEAR_DATA_REQ	0x34	Clear Data Request

1	data_to_clear	0x01 – channel reference table 0x02 – program label 0x04 – SIB 1 0x08 – SIB2 0x10 – Last channels selected 0x1F– all sections (except last channels selected)	user data to clear

BYTE	FIELD	VALUE	DESCRIPTION
0	CLEAR_DATA_CFM	0xB4	Clear Data Confirm
1	CMD RESULT	0x01 – cmd succesfull 0xFF – cmd fails	Command result. This command can fails because the byte 1 in the command buffer has a wrong number

3.3.39 PROGLABEL_LIST_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGLABEL_LIST_REQ	0x36	Program Label List Request

BYTE	FIELD	VALUE	DESCRIPTION
0	PROGLABEL_LIST _CFM	0xB6	Program Label List Confirm header.
1 - 2	size	0x0000 – 0xFFFF	Size of the Program Label List Confirm response. MSB in byte 1. LSB in byte 2.
3	program_type	1 - 31	The first available program type.
4 - 19	program_label	ISO-8859-1 character set.	program label.
……	……	……	……
……	……	……	……
……	……	……	……
size 17	program_type	1 - 31	The last available program type.
size 16 to size (1-2)	program_label	ISO-8859-1 character set.	program label.

3.3.40 BIC_LABEL_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	BIC_LABEL_REQ	0x37	BIC Label Request
1	location	0x00 – AUDIO 0x01 - DATA	Since SDEC can extract two services simultaneously, XM stack assigns an extraction location.
2	extartist_label_flag	0x00 – no request 0x01 – get artist label	If the system controller is requesting the extended BIC artist label, then set this flag to 0x01.
3	extsong_label_flag	0x00 – no request 0x01 – get song label	If the system controller is requesting the extended BIC song label, then set this flag to 0x01.

4	text_label_flag	0x00 – no request 0x01 – get BIC text	If the system controller is requesting the BIC text label, then set this flag to 0x01. **RECOMMENDATION: It's strongly recommended that the system controller set the text_label_flag to 0x00. For BIC text, its recommended system controller use BIC_TEXT_REQUEST command.**

BYTE	FIELD	VALUE	DESCRIPTION
0	BIC_LABEL_CFM	0xB7	BIC Label Confirm header.
1	command_status	0x01 - SUCCESFUL 0xFF - FAILURE	Command status
2	location	0x00 - location 0 0x01- location 1	Since SDEC can extract two services simultaneously, XM stack assigns an extraction location
3	sid	1 - 255	SID of extracted location.
4	extartist_flag	0x00 – ABSENT 0x01 – PRESENT	artist label present or absent from response. Please note this octet is only present when the extartist_label_flag is set to 0x01 (PRESENT) in BIC_LABEL_REQ command.
5 to 6	extartist_size	1 to 80	artist label size Please note this byte is only present when the extartist_label_flag is set to 0x01 (PRESENT). in BIC_LABEL_REQ command.
7 to (6+exartist_size)	extartist_label	ISO-8859-1 character set	artist name Please note this byte is only present when the extartist_label_flag is set to 0x01 (PRESENT).in BIC_LABEL_REQ command Note: The maximum size of the BIC extended artist name is 80 bytes. Therefore artist_size < = 80 bytes.
artist_size +8	extsong_flag	0x00 – ABSENT 0x01 – PRESENT	song label present or absent from response Please note this octet is only present when the extsong_label_flag is set to 0x01 (PRESENT) in BIC_LABEL_REQ command.
artist_size + 9 to artist_size +10	extsong_size	1 - 80	song label size. Please note this byte is only present when the extsong_label_flag is set to 0x01 (PRESENT).in BIC_LABEL_REQ command.

artist_size + 11 to artist_size +10	extsong label	ISO-8859-1 character set	song label Please note this byte is only present when the extsong_label_flag is set to 0x01 (PRESENT) (PRESENT).in BIC_LABEL_REQ command Note: The maximum size of the BIC extended song title is 80 bytes. Therefore song_size < = 80 bytes.
artist_size +song_size + 12	bic_text_status	0x00 – ABSENT 0x01 – PRESENT	If BIC text was not received, then DSP returns ABSENT. If BIC text was received, then DSP returns PRESENT.Please note this octet is only present when the text_label_flag is set to 0x01 (PRESENT) in BIC_LABEL_REQ command
artist_size + 13 to artist_size + 14	text_size	1 - 253	text size. Please note this byte is only present when the text_label_flag is set to 0x01 (PRESENT). in BIC_LABEL_REQ command
artist_size + 15	sid	1 - 255	SID of extracted location. Please note this byte is only present when the text_label_flag is set to 0x01 (PRESENT) in BIC_LABEL_REQ command
artist_size + song_size +16	scroll_narrowcast	see specification	Scrolling and narrowcast information. Please note this octet is only present when the text_label_flag is set to 0x01 (PRESENT) in BIC_LABEL_REQ command. Reference: Service Layer Specification; Revision 1.4; Section 8.11
artist_size + song_size +17	text_size	1-253	size of the text message Please note this octet is only present when the text_label_flag is set to 0x01 (PRESENT) in BIC_LABEL_REQ command.
artist_size + song_size +18 to artist_size + song_size + text_size + 17	scroll_narrowcast	see specification	BIC Text. Please note this octet is only present when the text_label_flag is set to 0x01 (PRESENT) in BIC_LABEL_REQ command. Reference: Service Layer Specification; Revision 1.4; Section 8.11

3.3.41 DISPLAY_MASK_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	DISPLAY_MASK_REQ	0x4C	Display Mask Request
1	service_label	8, 10, 16	service label display mask
2	artist_label	8, 10, 16	artist label display mask
3	song_label	8, 10, 16	song label display mask
4	program_label	8, 10, 16	program label display mask

3.3.42 SDEC_VER_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	SDEC_VER_REQ	0x4D	Read Data Request
1	sdec_version	0x21 HCMOS 7 0x25 HCMOS 8	SDEC version number expected by system controller.

3.3.43 FAST_CLEAR_EVENT

BYTE	FIELD	VALUE	DESCRIPTION
0	FAST_EVENT_CLR_REQ	0x50	Event or error bit/s clear request
1	Reg	0x45 – 0x48 0x4B – 0x4C	Event or Error register to be written
2	mask		Mask to apply to the specified register (e.g. 0x01 clear bit 1 of the specified register)

3.3.44 FAST_DISABLE_INT_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	FAST_DISABLE_INT_REQ	0x51	Event or error interrupt/s disable request
1	Reg	0x41 – 0x44 0x49 – 0x4A	Event and Error enable interrupt register to be written
2	mask		Mask to apply to the specified register (e.g. 0x01 clear bit 1 of the specified register)

3.3.45 FAST_ENABLE_INT_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	FAST_ENABLE_INT_REQ	0x52	Event or error interrupt/s enable request
1	Reg	0x41 – 0x44 0x49 – 0x4A	Event and Error enable interrupt register to be written
2	Mask		Mask to apply to the specified register (e.g. 0x01 set bit 1 of the specified register)

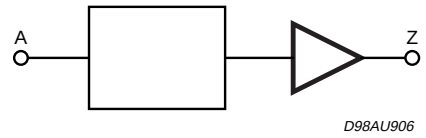
3.3.46 FAST_EXTRACT_REQ

BYTE	FIELD	VALUE	DESCRIPTION
0	FAST_EXTRACT_REQ	0x53	Fast Extract Request header
1	Location	0x00 – location 0 0x01 – location 1	Since DSP can extract two services simultaneously, XM stack assigns an extraction location.
2	channel_or_sid_flag	0x00 – use channel number given in byte 3. 0x01 – use SID given in byte 4	If channel number is used, then the SDEC will get the SID and PCIDs from CRW.

3	channel_number	1 - 255	channel number
4	Sid	1 – 255	Service Identifier
5	Extract_type	0x03 AMBE 0x05 AAC 0x0A Trans_data	Service Component Type to extract.
6	primary_pcid	5 – 254	Primary Payload Channel Identification. SDEC will check the PCID given by the system controller is still the same as the one in CRW, If different, then the extraction fails and SDEC sets the FAST_EXTRACTION_COMPLETEDx event bit. Reference: Transport Layer Specification; Revision 1.2; page 40
7	secondary_pcid	5 – 254	Secondary Payload Channel Identification SDEC will check the PCID given by the system controller is still the same as the one in CRW, If different, then the extraction fails and SDEC sets the FAST_EXTRACTION_COMPLETEDx event bit. Reference: Transport Layer Specification; Revision 1.2; page 40
8	Routing	0x00 – No routing 0x01 – AUD_PORT 0x02 – DAT_PORT 0x03 – BOTH_PORT	Routing information A data service component can only be routed to the data port (DAT_PORT). An audio service component is routed to audio port (AUD_PORT), data port (DAT_PORT) or both the audio and data port (BOTH_PORT).

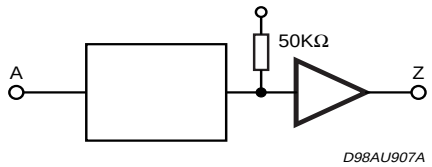
4.0 I/O CELLS DESCRIPTION)

Input Pad Buffer



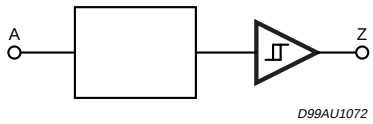
EXTERNAL PIN	INPUT CAP.
A	1.3pF

2) Input Pad Buffer with Active Pull-up



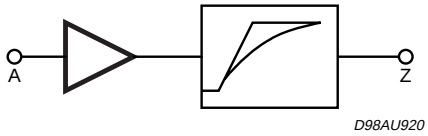
EXTERNAL PIN	INPUT CAP.
A	1.3pF

3) Schmitt Trigger Input Pad Buffer



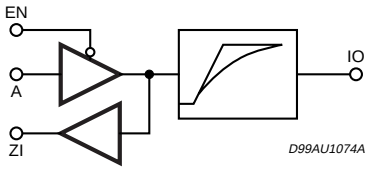
EXTERNAL PIN	INPUT CAP.
A	1.3pF

4) Output Pad Buffer, 4mA with slew rate control**.**



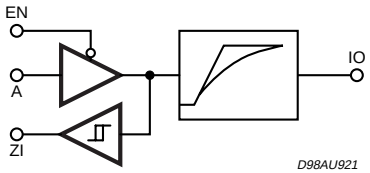
EXTERNAL PIN	MAX LOAD
A	100pF

5) BiDir Pad Buffer, 4mA with slew rate control.



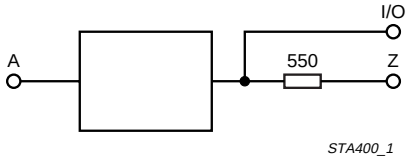
EXTERNAL PIN	INPUT CAP.	MAX LOAD
I/O	1.3pF	100pF

6) Schmitt Trigger BiDir Pad Buffer, 4mA with slew rate control.



EXTERNAL PIN	INPUT CAP.	MAX LOAD
I/O	1.3pF	100pF

7) Analog Pad Buffer

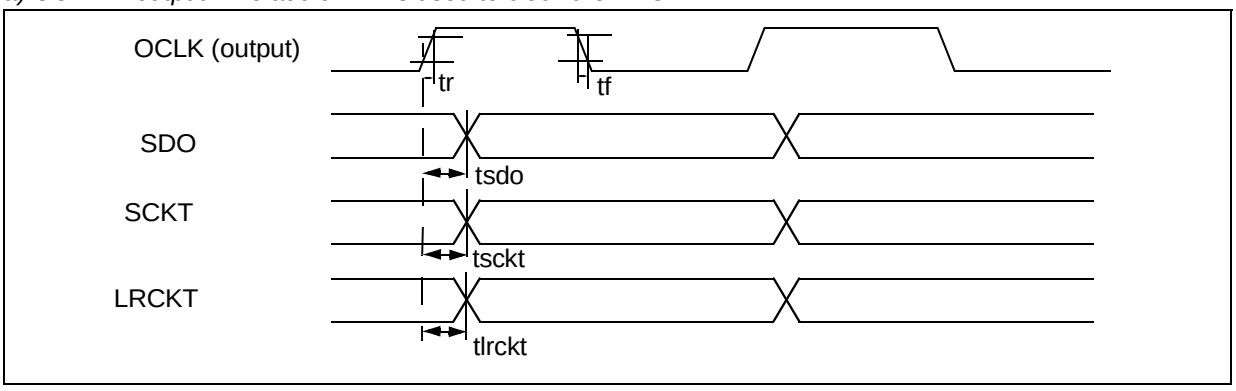


EXTERNAL PIN	INPUT CAP.
A	1.9pF

5.0 TIMING DIAGRAM (GUARANTED BY DESIGN)

5.1 Audio DAC Interface (PCM)

a) OCLK in output. The audio PLL is used to clock the DAC



Notations: Cload_XXX is the load in pF on the XXX output. pad_timing(Cload_XXX) is the propagation delay added to the XXX pad due to the load. The following table gives this timing

.

Load (pF)	max pad_timing (rise)	max pad_timing (fall)
25	6 ns	10.6 ns
50	10 ns	13 ns
75	12 ns	15.4 ns
100	14 ns	18 ns

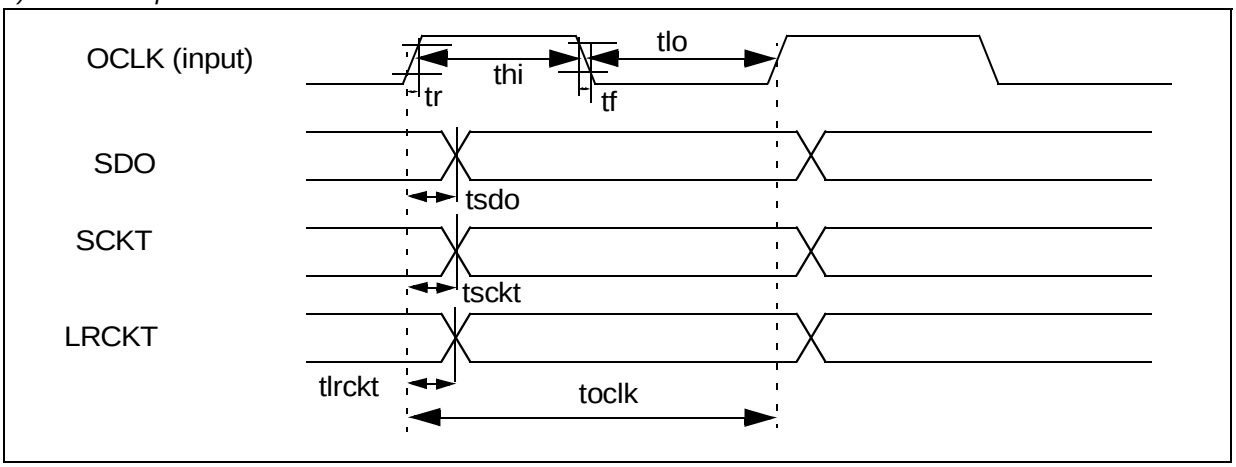
tsdo_max = 3.5 + pad_timing(Cload_SDO) - pad_timing(Cload_OCLK)

tsckt_max = 4.0 + pad_timing(Cload_SCKT) - pad_timing(Cload_OCLK)

tlrckt_max = 3.5 + pad_timing(Cload_LRCKT) - pad_timing(Cload_OCLK)

The Tr and Tf depends on the load on the different pads but at a given load, the Tr and Tf timings are identical for OCLK(output case), SCKT, LRCKT, SDO.

b) OCLK in input



Thi min = 5 ns

Tlo min = 5 ns

Toclk min = 25 ns

tsdo_max = 5.5 + pad_timing(Cload_SDO) ns

tsckt_max = 6.0 + pad_timing(Cload_SCKT) ns

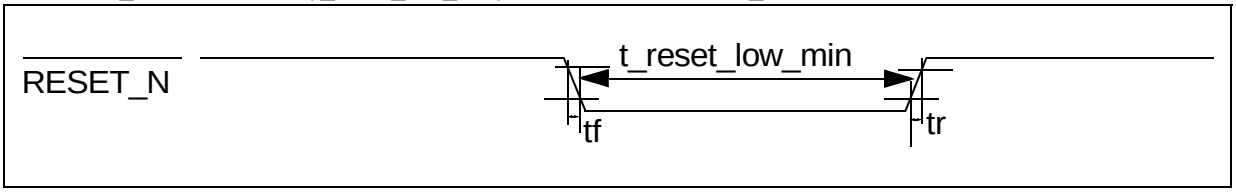
tlrckt_max = 5.5 + pad_timing(Cload_LRCKT) ns

tr max = 20 ns

tf max = 20 ns

5.2 RESET_N

The Reset_N min duration (t_reset_low_min) is 10 us. The RESET_N is active low.

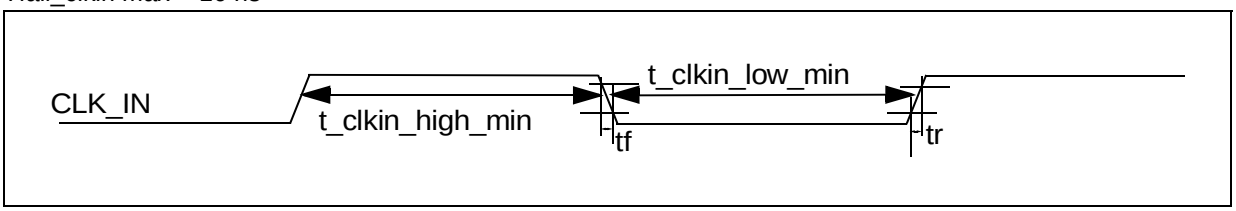


5.3 CLK_IN

The CLK_IN typical frequency is 23.92 Mhz. The min high and low time are 5 ns.

Trise_clkin max = 16 ns

Tfall_clkin max = 16 ns

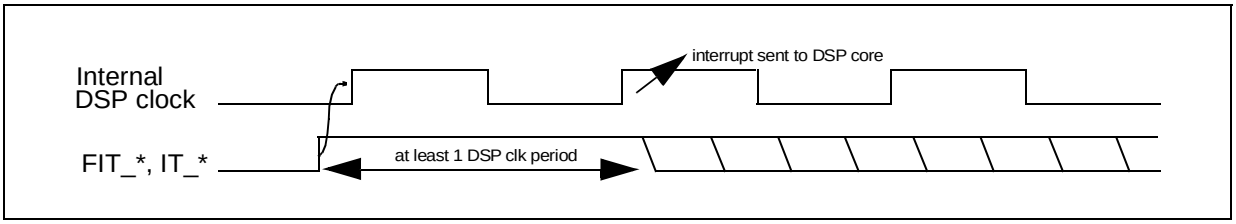


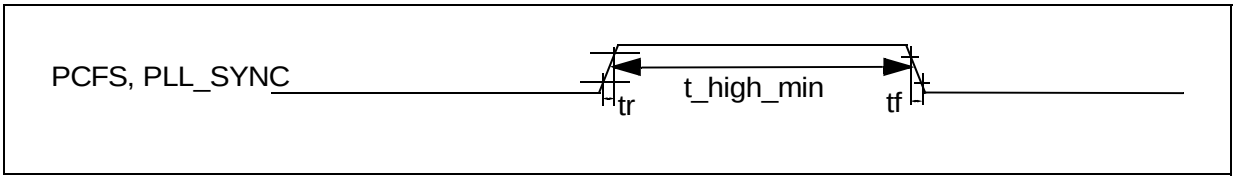
5.4 Single bit inputs : PCFS, PLL_SYNC

The min pulse duration (high or low) on these inputs is 20 ns.

Trise max = 20 ns

Tfall max = 20 ns

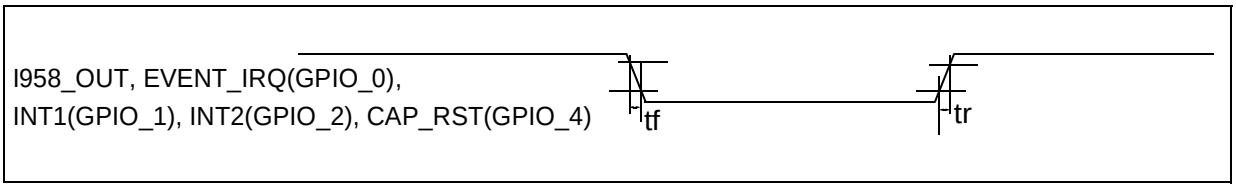




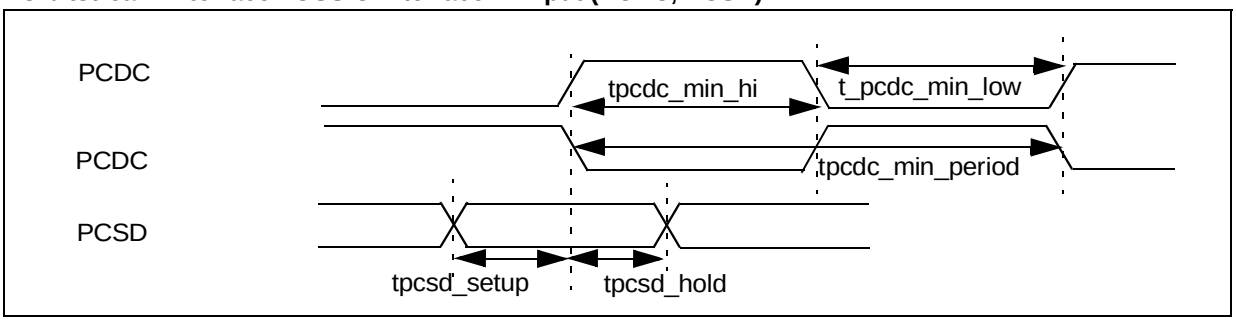
5.5 Single bit outputs : I958_OUT, EVENT_IRQ(GPIO_0), INT1(GPIO_1), INT2(GPIO_2),

CAP_RST(GPIO_4)

Load (pF)	max pad_timing (rise)	max pad_timing (fall)
25	6 ns	10.6 ns
50	10 ns	13 ns
75	12 ns	15.4 ns
100	14 ns	18 ns



PC bitstream interface : USSIO interface in input (PCDC, PCSD)



tpcsd_setup_min = 15 ns

tpcsd_hold_min = 15 ns

tpcdc_min_hi = 20 ns

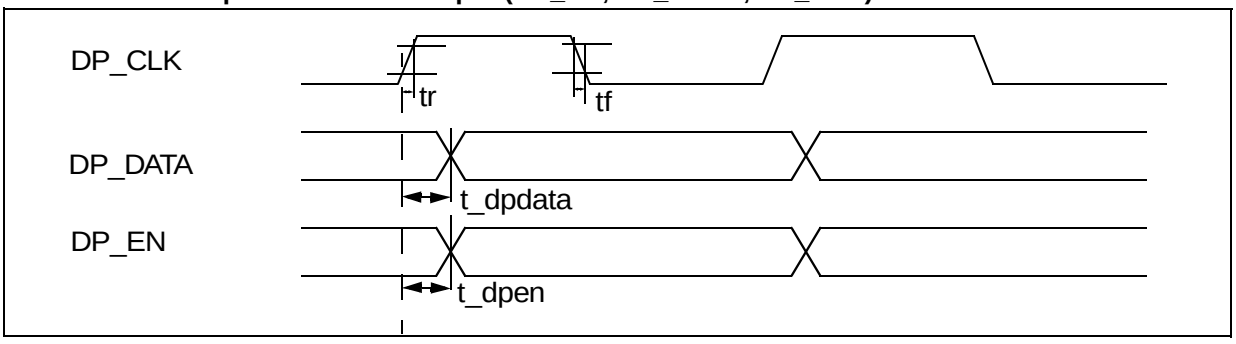
tpcdc_min_low = 20 ns

tpcdc_min_period = 60 ns

tr_max = 20 ns

tf_max = 20 ns

5.6 USSIO in output : Data Port output (DP_EN, DP_DATA, DP_CLK)



Notations: Cload_XXX is the load in pF on the XXX output. pad_timing(Cload_XXX) is the propagation delay added to the XXX pad due to the load. The following table gives this timing.

t_dpen_max = 6.0 + pad_timing(Cload_DP_EN) - pad_timing(Cload_DP_CLK)

Load (pF)	max pad_timing (rise)	max pad_timing (fall)
25	6 ns	10.6 ns
50	10 ns	13 ns
75	12 ns	15.4 ns
100	14 ns	18 ns

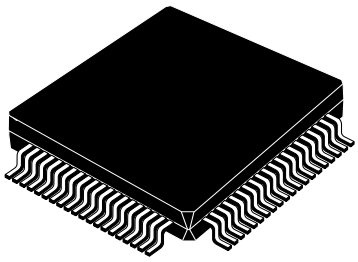
t_dpdata_max = 6.0 + pad_timing(Cload_DP_DATA) - pad_timing(Cload_DP_CLK)

The Tr and Tf depends on the load on the different pads but at a given load, the Tr and Tf timings are identical for DP_CLK, DP_DATA, DP_EN.

DP_DATA and DP_EN must be sampled on the falling edge of DP_CLK.

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.22	0.32	0.38	0.009	0.013	0.015
C	0.09		0.20	0.003		0.008
D		16.00			0.630	
D1		14.00			0.551	
D3		12.35			0.295	
e		0.65			0.0256	
E		16.00			0.630	
E1		14.00			0.551	
E3		12.35			0.486	
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.0393	
K	3.5°(min.), 7°(max.)					

OUTLINE AND MECHANICAL DATA



TQFP80 (14x14x1.40mm)